



# Case Management

17 October 2023



# CONTENTS

Case management.....	5
Introduction to case management.....	6
Identifying case types elements.....	9
Understanding case hierarchy.....	14
Identifying people involved in case types.....	17
Building case types.....	20
Providing data for cases.....	24
Understanding and complying with case management best practices.....	26
Automating work by creating case types.....	27
Case types.....	28
Case life cycle elements.....	43
Configuring a data model for a case.....	150
Views for cases.....	168
Remote case types.....	188
Configuring and tailoring case types.....	203
Configuring case type settings.....	203
Configuring complex processes.....	248
Managing case types.....	315
Customizing the Get Next Work logic.....	317
Changing case statuses.....	318
Completing work on time.....	324
Urgency.....	326
Setting service-level agreements (SLAs) for case resolution.....	329
Setting service-level agreements (SLAs) for stages, processes, and steps.....	330
Adjusting initial urgency for an assignment.....	332
Creating a service-level agreement (SLA) rule.....	332

Configuring delayed service-level processing.....	334
Defining an escalation action for a case.....	335
Defining an escalation action for a stage and a process.....	336
Defining an escalation action for an incomplete assignment.....	337
Engaging and notifying stakeholders.....	338
Creating a notification.....	339
Defining a notification message.....	339
Adding recipients for a notification.....	340
Notification channels.....	341
Empowering knowledge workers.....	348
Responding to business exceptions in a flow.....	348
Configuring and working with optional actions in case types.....	353
Preparing for collaboration with users by using Pulse.....	360
Integrating cases with external resources.....	374
Integrating a case with an external resource.....	374
Directed Web Access in configuring assignments for external users.....	387
Enabling cases for creation through web and chat channels.....	392
Sourcing attachments from external storage.....	392
Storing case attachments using external storage.....	400
Designing questionnaires.....	405
Creating a questionnaire.....	406
Running a questionnaire in a case.....	424
Fine-tuning your questionnaire.....	426
Visualizing questionnaire data by creating insights in Cosmos React.....	435
Using advanced questionnaire features in Dev Studio.....	436
Securing case access.....	453
Configuring case type access.....	453
Controlling access to case attachments.....	457
Interacting with cases at run time.....	463

Creating and managing cases.....	463
Creating tools for facilitated work processing.....	491
Collaborating on cases.....	499

# Case management

Model, track, and manage your business processes with Pega Platform™ case management. By creating a visual representation of your business cases and grouping related tasks into stages, you can easily monitor how and when the tasks are completed.

A case, which represents a business process, is made up of many stages, processes, tasks, policies, and supporting content. The case as a whole continues to change throughout its life cycle due to internal and external events. Depending on the context of the case, individual tasks, processes, or stages can be resolved by different customer service representatives (CSRs). This flexibility helps you achieve your goals in the most effective way.

Processing cases with case management increases efficiency, consistency, and the visibility of case processes, which decreases costs and improves quality. Accordingly, case management supports the way that CSRs work with case timelines and relevant context, such as documents and attachments. Case management facilitates resolving business processes, and guiding CSRs to the information and tasks most relevant to their roles in the case. With personalized event notifications, spaces, and Pulse, CSRs can collaborate, ensuring that every case worker is well-informed about a progress in a case.

The following figure shows a visual representation of a business process that can result in approving or rejecting a job

candidate:



A business process of reviewing job candidates presented in Case Designer

# Introduction to case management

Use case management to deliver goal-oriented results by handling business cases from start to resolution and combining human actions with digital automations. In case management, you visualize your business process by including people, data, and actions, so that you can prepare a flexible path that leads to your objective. For example, you can model a process of approving candidates after reviewing job applications, from collecting documents from a candidate, through the job interview, to the final approval. Apart from designing the most common path towards resolution, in case management you can dynamically adjust your work to respond to changing circumstances. For example, your business process might include an additional job interview when this action is relevant to the current situation. With case management, you can successfully resolve business processes that follow an unclear or unpredictable workflow.

Pega Platform provides tools to implement case management as the way to meet your business objectives. You can create goal-oriented solutions by defining reusable templates for business processes that guide users to a successful resolution, but also provide flexibility for an appropriate reaction in changing circumstances. Make right decisions about the best course of action that leads to your defined objective by taking advantage of the following aspects of Pega case management:

## **Business process visualization**

Users can better understand and analyze a business process through visualization. After creating a visual representation of the workflow, people, and data that the business process involves, users can follow the most efficient path to their business objective. Pega Platform provides an intuitive editor to define your business process in the way that reflects work that you need to perform. Through visualization, you can transparently communicate your business process to others, for example to stakeholders.

## **Goal-oriented business outcome**

Users can focus on the crucial part in a business process, that is, reaching a specific goal. Users meet their business objectives by following a predefined path or by adjusting the sequence of actions to dynamically changing circumstances.

## **Support for non-linear business activities**

Users can flexibly start supporting business processes that are most relevant to the current circumstances.

## **Ad hoc tasks**

Users can go beyond the predefined order of assignments by creating dynamic tasks and assigning them to individuals. Adding instructions and setting goals and deadlines facilitates the assignment resolution.

## **Enhanced business flexibility**



Users can move a business process to any passed or subsequent milestone for additional processing to ensure that the work is fully completed.

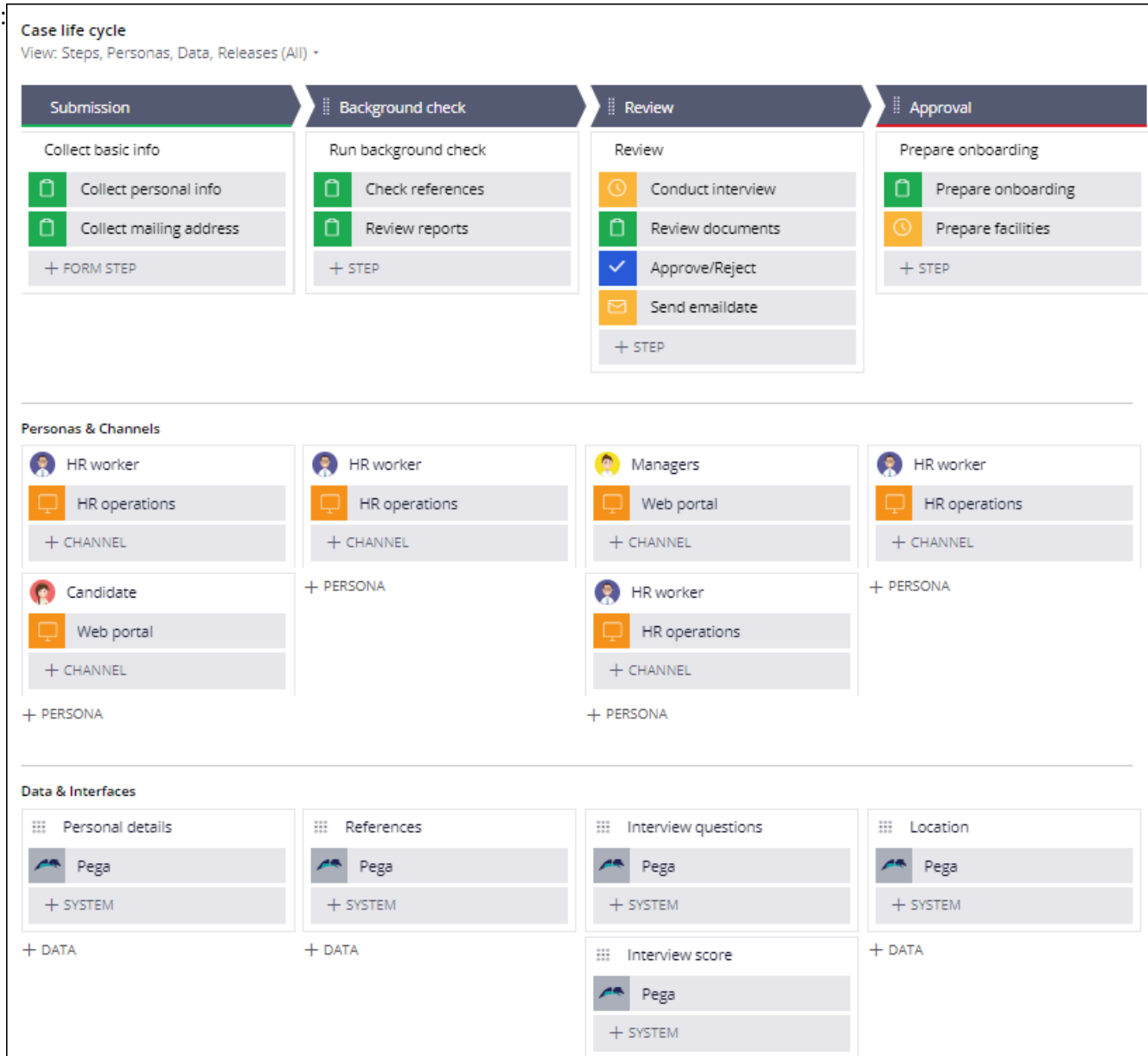
**Transparency of information**

Users have a clear view of how a business process reaches a certain point, even after re-routing and additional processing, by following an audit trail and analyzing changes of individual fields and values.

**People-driven and collaboration-centric processes**

Users can share progress and changes in business processes by using multiple collaboration tools, such as instant messages or documents, to speed up resolution and ensure that everyone has relevant information. To keep users well-informed, Pega Platform also offers notifications about various events in the workflow.

The following figure shows a sample visualization of a business process with a goal of approving a job candidate:



*Visualization of a process of approving a job candidate*

your business circumstances change, you can add and remove steps in the process, add optional and supporting actions, and, finally, manage data and personas in the process.

Pega Platform conforms to the Business Process Model and Notation (BPMN) standard by using a subset of BPMN, but also extends the standard with smart shapes such as the Capture signature, Cascading approval, and Wait shape. The flow diagram includes a smart shapes palette that provides a set of frequently used, pre-configured utilities and sub-processes. The shapes in the palette can be used in process flows with little or no additional configuration beyond what BPMN supports. Pega continues to define and redefine the BPM space by favoring a more robust approach that supports advanced enterprise BPM constructs, such as business rules and case management (for example, treating GUI calls as a service instead of as an intrinsic part of the BPM flow).



**What to do next:** Before you start delivering end-to-end digital solutions for your customers, learn how you can fully benefit from Pega Platform case management:

### Identifying case types elements

To maximize efficiency while processing work, analyze your business processes, needs, and requirements. After you identify and evaluate the main elements of your work, you can take full advantage of the possibilities that Pega Platform provides to define and visualize your business cases by creating case types.

In Pega Platform, you resolve your business processes through case management. You apply this method when you can clearly define the goal that you want to achieve, but the course of your business process is unclear or changeable. In Pega Platform you can use tools that help you achieve your business objectives in a flexible way, so that you can respond to dynamic and unique circumstances.

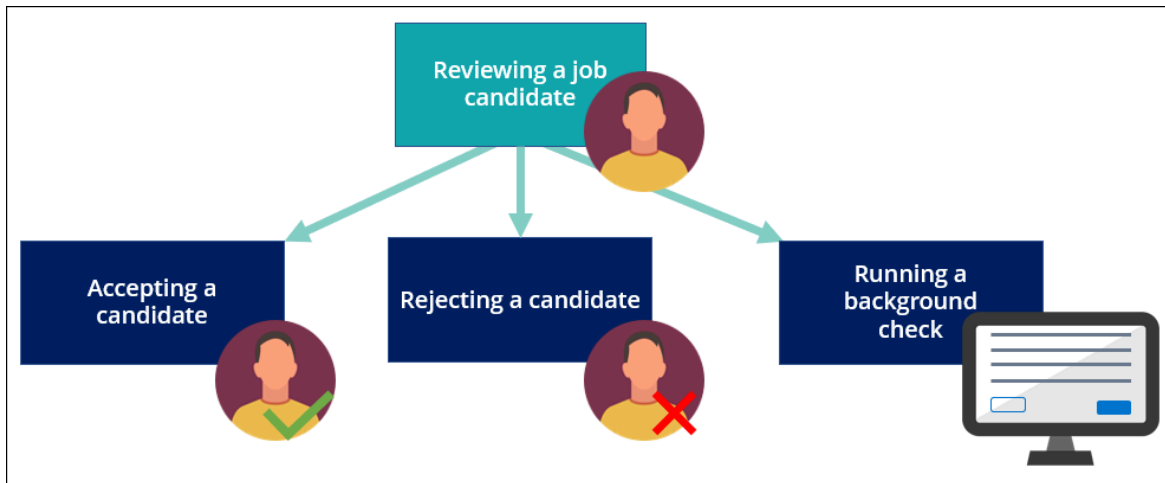
A case type is a visual representation of your business process; a template for work that you can reuse for processing multiple instances of the same business case, such as reviewing applications from job candidates. By creating a case type, you define the path that your work follows, the people who are involved in processing a case, and the data that the case requires. As a result, you save time and ensure that your business processes can reach a successful resolution.

Pega Platform offers multiple tools and possibilities so that you can design your case types to meet your unique business needs. Before you start using Pega Platform to process your work, think about the following factors:

#### The outcome that you want to achieve

Think about the result that you want to achieve with your work. Consider whether there might be more than one outcome to a business process. For example, if your objective is to review job candidates, think about the possible resolutions: you might accept or reject the candidate, or you might want to redo or redirect the process before reaching a conclusion.

The following figure shows a sample business process that might result in three different outcomes:

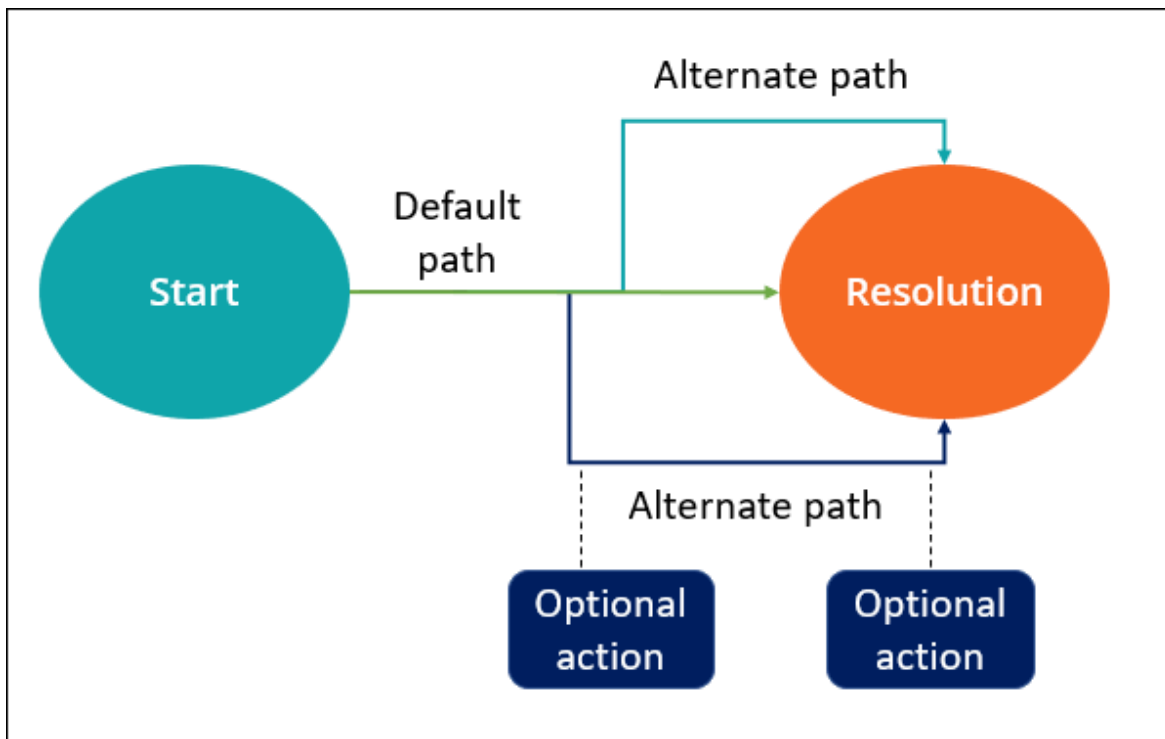


Outcomes of a case type

### The paths that lead to the outcome

While planning your business process, think about all the possible paths that the process might take before reaching the outcome. One of the possibilities is a default path without any exceptions or errors. However, consider any points in the process that might cause delays or alternations from this default path, and then think about solutions that can bring your work to resolution under new circumstances. Design alternate paths that lead to a resolution when the default path is unable to do so, such as an alternative for when you reject a candidate during a job application review. By creating alternate paths, you respond to dynamic situations in an agile and flexible way. You can also add actions to the business process to increase the chances of reaching the expected outcome. For example, you can apply service-level agreements to actions that users perform, so that the work can advance in a timely manner.

The following figure shows a graph of a sample business process with a default path and two alternate paths. Additionally, one alternate path includes two optional actions:

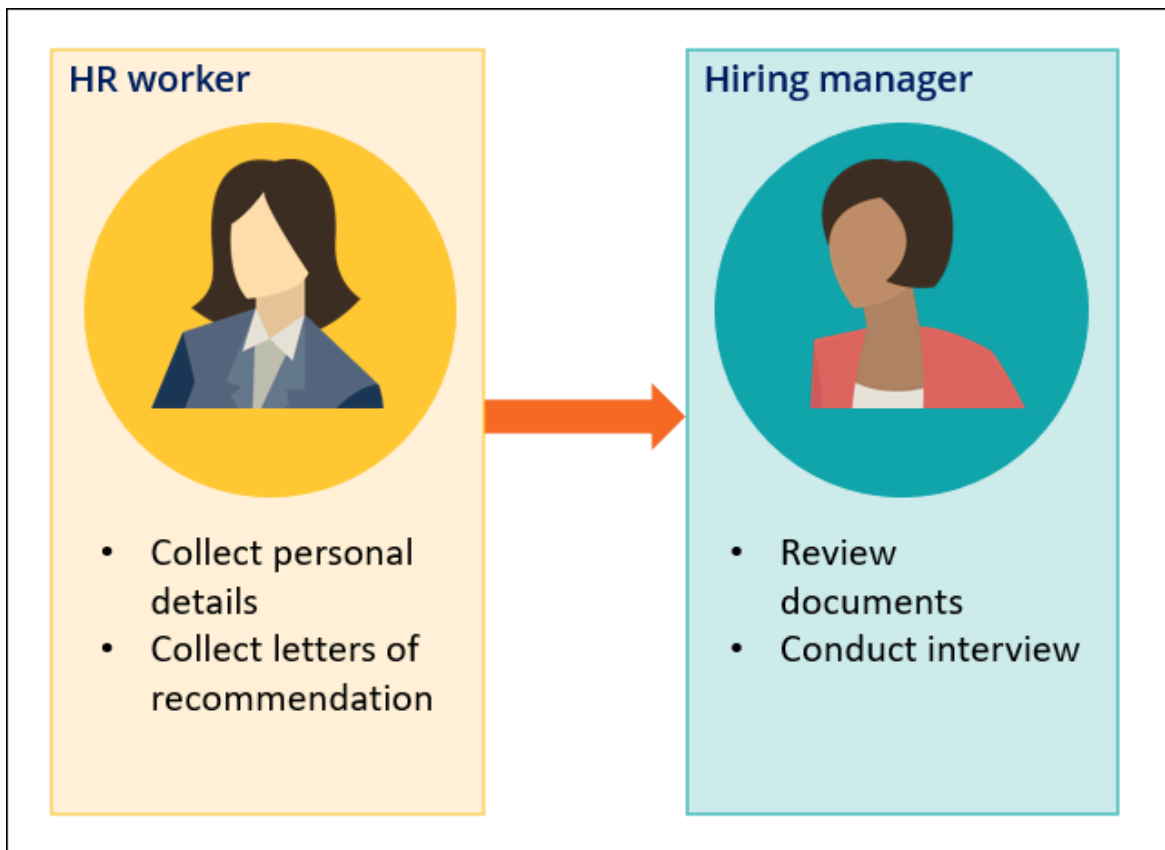


Paths towards resolution

### The elements of your business process

To divide your business process into smaller elements, think about the milestones or turning points that your work involves. For a milestone, consider the transitions in a process from one person to another, or crucial events on the business process path. Consider the actions that people need to take at each milestone – whether you can logically group the actions or perform multiple sets of actions in parallel, who performs the actions, and whether you can automate any of the actions. For example, in a job application review, a milestone might be the transition of work from an HR worker to a hiring manager, after the HR worker completes the first set of actions that might be collecting an applicant's personal details, letters of recommendation, and proof of education. By using Pega Platform, you increase flexibility of your applications because you can seamlessly move between the actions in an order that works best for your current business needs.

The following figure shows a transition of work from an HR worker to a hiring manager:

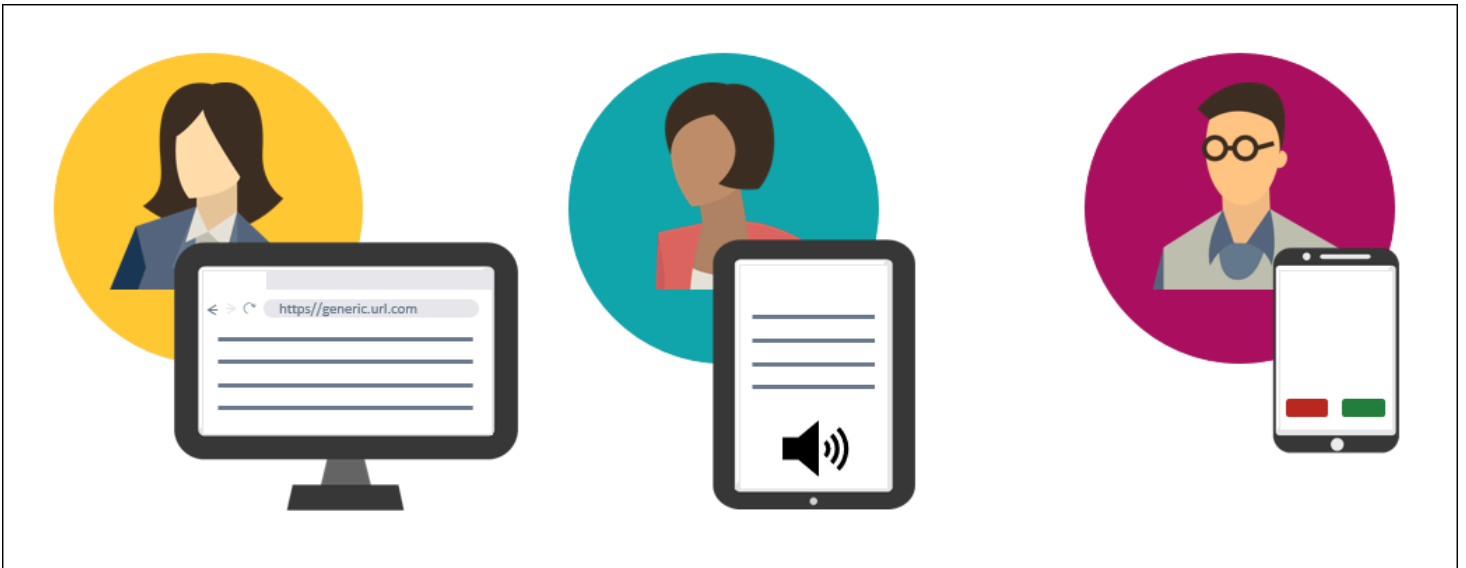


Transition of work

### The people involved in your business process

When defining your work, think about the people needed for your case to reach its outcome, and the channels that they might use while interacting with your business process. Consider the different means of communication that you can apply, such as email and push notifications, as well as the devices that case participants might work with, such as mobile phones and tablets. Define the elements that particular participants need to access so that you can avoid exposing users of your application to irrelevant content. Finally, decide whether your application users need both online and offline access to perform their work.

The following figure shows three users with different devices that they use to process work: a personal computer, a tablet with a screen reader, and a mobile phone:

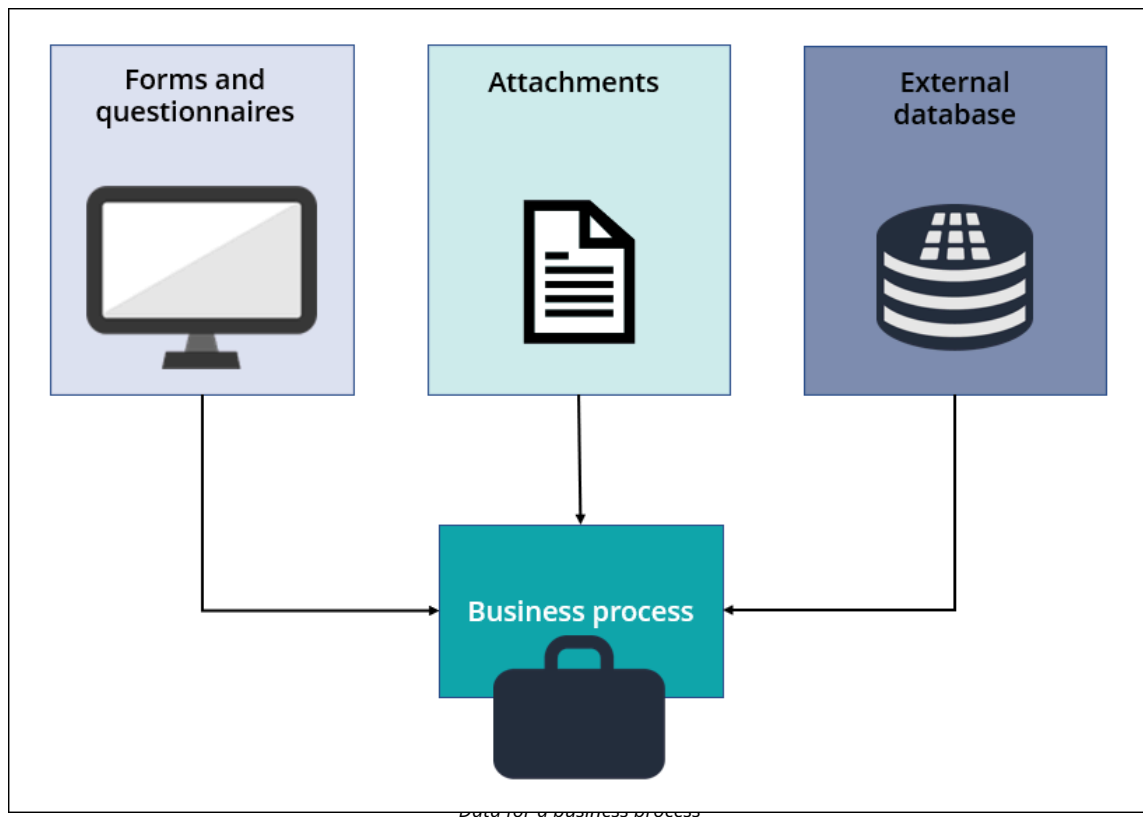


*Different types of users and devices*

### **The data that your business process requires**

Define the data that your business process needs to reach a resolution, and think about the different ways of providing the information. Consider what information users provide, and the best way of collecting that data – a form with fields to complete, a survey, or a document attached to a business process. Also, think about the information that you need to fetch from external systems or databases, and the integrations that your application might require.

The following figure shows a business process that sources data from forms and questionnaires, attachments, and an external data base:



**What to do next:** After you identify and analyze the elements that you need to create a case type, think about interactions and dependencies between case types. See [Understanding case hierarchy \(on page 14\)](#).

### Related information

[App Studio overview \(on page 14\)](#)

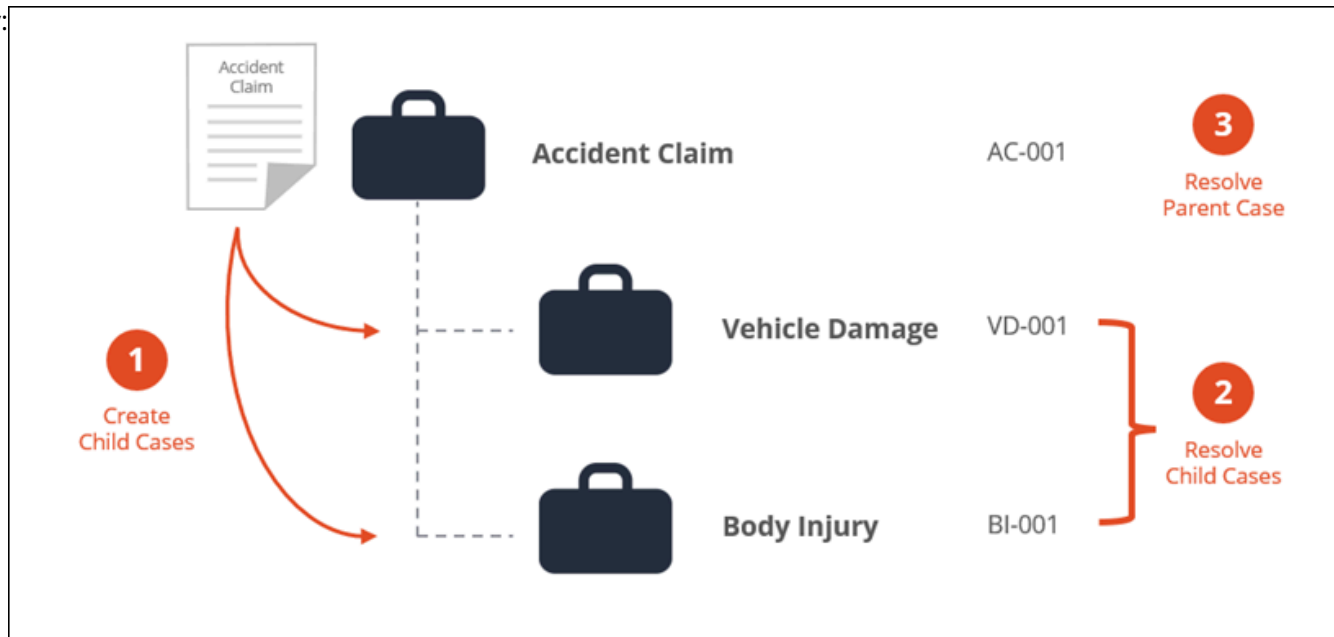
[Creating a Microjourney for customer success \(on page 14\)](#)

### Understanding case hierarchy

Before you start visualizing your business cases, analyze the relationships and interactions between them. When you think about the outcome that you want to achieve, decide whether you need to create only one template to complete your work, or several templates for smaller results that add up to your ultimate goal. Analysis of your business needs helps you make the most of the tools that Pega Platform provides and create the most efficient solutions.

For example, if your business objective is to review insurance claims, you might need to consider different scenarios while processing a claim. In a car accident insurance claim, a customer might file both body injury and vehicle damage claims. To process the case, and to estimate the total amount of money that the customer should receive in compensation, you need to process the injury and damage cases first. Using the results of these two cases, you can calculate the

final insurance payout. To achieve this goal, you can create an Accident Claim parent case, and then Body Injury and Vehicle Damage child cases. You resolve the parent case only after all the child cases reach a resolution. Pega Platform refers to the relationship between parent and child case types as the case type hierarchy. The following figure presents a same case type hierarchy:



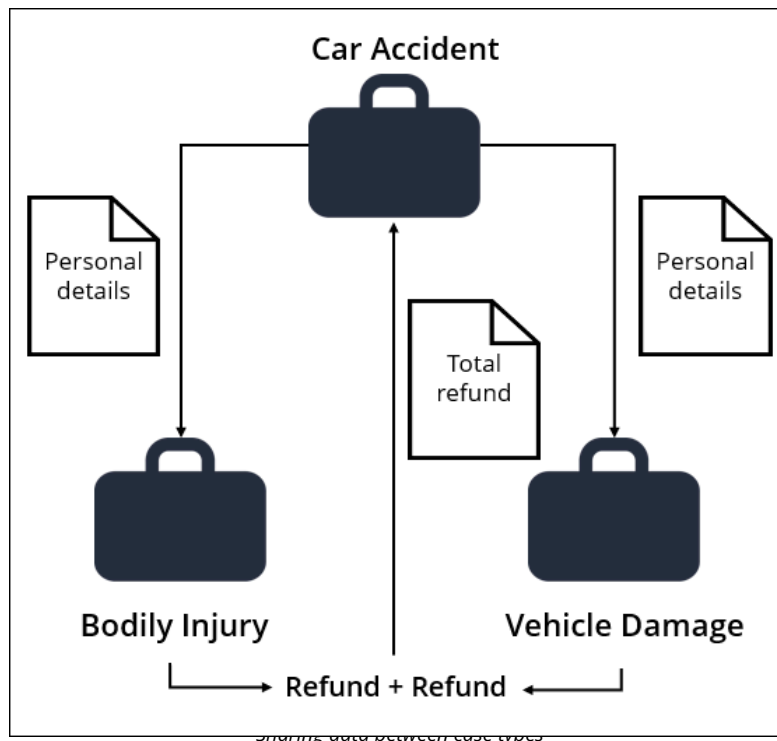
*Relationships between case types*

Making your work more granular also results in time and costs savings, because you can reuse child cases in different scenarios. For instance, in a claims process for a customer that files a car accident insurance claim that requires only a Body Injury child case, you can reuse your existing case types instead of creating a whole new business process. Reusing child case types increases the flexibility of your application, and helps you meet unique and dynamically changing needs. By using child case types as building blocks, you can resolve complex business cases and dependencies without planning and creating long, complicated case types.

## Sharing data between cases

For greater automation of your business processes, you can configure how parent and child cases share data. Each time you create a child case, your application can populate it with specific information from the parent case, so that you do not need to enter the same data twice. For example, in a car accident insurance claim, a Bodily Injury child case can include all of the personal details that a customer provides for the parent case. A parent case can also reflect any values that you enter while processing its child cases. For instance, when you decide how much money a customer should receive from Bodily Injury and Vehicle Damage child cases, the amount can be automatically totaled and added to the Car Accident parent case.

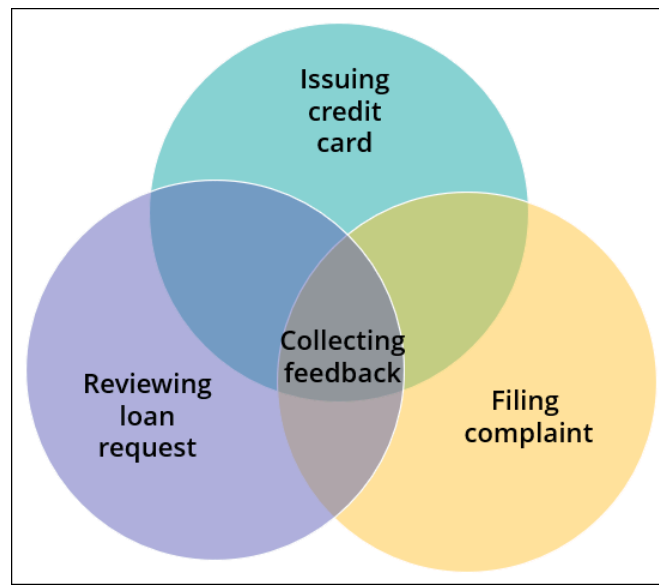
The following figure shows a graph that illustrates sharing data between parent and child case types:



## Reusing child case types in different scenarios

When you want to define a reusable part of a business process, consider a set of tasks that you can complete independently of your main business process, and that you can apply in different scenarios. Analyze a sample scenario in which a bank reviews various requests from customers. The requests might include issuing a duplicate credit card, filing in a complaint, and making a loan request. After a customer service representative (CSR) resolves each request, a customer receives a short questionnaire to collect feedback. Collecting feedback is a separate process that includes creating a questionnaire, sending an email to a customer, sending a reminder after a week without a reply, and then analyzing the customer's feedback. The collecting feedback process can be the same for many other business processes, as shown in the following figure:





By reusing child case types in different scenarios, you speed up application development and provide a consistent experience for your customers.

**What to do next:** After you understand dependencies between your business processes, identify types of users that you need to involve in your cases. See [Identifying people involved in case types \(on page 17\)](#).

For relevant training materials, see the [Creating a child case](#) module on Pega Academy.

### Related information

[Case types \(on page 28\)](#)

[Creating a top-level case type \(on page 30\)](#)

[Creating a child case type \(on page 31\)](#)

[Sharing data between parent and child cases \(on page 217\)](#)

[Creating a child case](#)

### Identifying people involved in case types

Along with actions and data, people involved in your business process are a key element in achieving your goals. Before you start modeling and automating your work with the tools that Pega Platform offers, define the people that you need to engage in your business process, what actions they need to perform, and what device they might use to interact with your application.

When you define the participants in your business process, answering the following questions might be helpful:

## Who do I need to involve?

Before you model your business process, think about the types of users that you intend to reach with your application. To determine what user types you need to create, consider what the specific goals and actions are for each type. For example, in a loan request review process, a customer service representative (CSR) has a different objective and tasks from a manager. While a goal for a CSR could be to collect income information from a customer, a manager might need to either approve or reject the customer's request. However, all CSRs that perform work in your application share one set of tasks, and as a result, form one user type. Pega Platform refers to user types as personas. By defining personas, you can analyze and determine what kind of information the members of each user type need to access, and what actions they need to complete. Consequently, you ensure that you build and configure an application that is relevant to all of the types of users involved in your business process.

## What actions do users need to perform?

When you identify personas in your business process, analyze what actions each persona might need to perform. For example, in a loan request process, a CSR might need to review the customer's creditworthiness and income information, and a manager might need to review that information and then approve or reject the loan request. At this point, you can also identify the data that these users need to complete their tasks. Consider also how user actions interact with each other. For example, how the manager's decision affects the process, or how your application needs to assign tasks to users. After you define all of the actions in your business process, you can use Pega Platform to visualize, model, and automate your work. As a result, you can create an application that precisely reflects your business process. Pega Platform also offers tools that you can use to dynamically adjust actions to your changing requirements. For example, a user might perform an action only when it is relevant to a specific case, such as a CSR who reviews a loan request only if a customer's income is higher than a specified amount.

## How will users interact with my application?

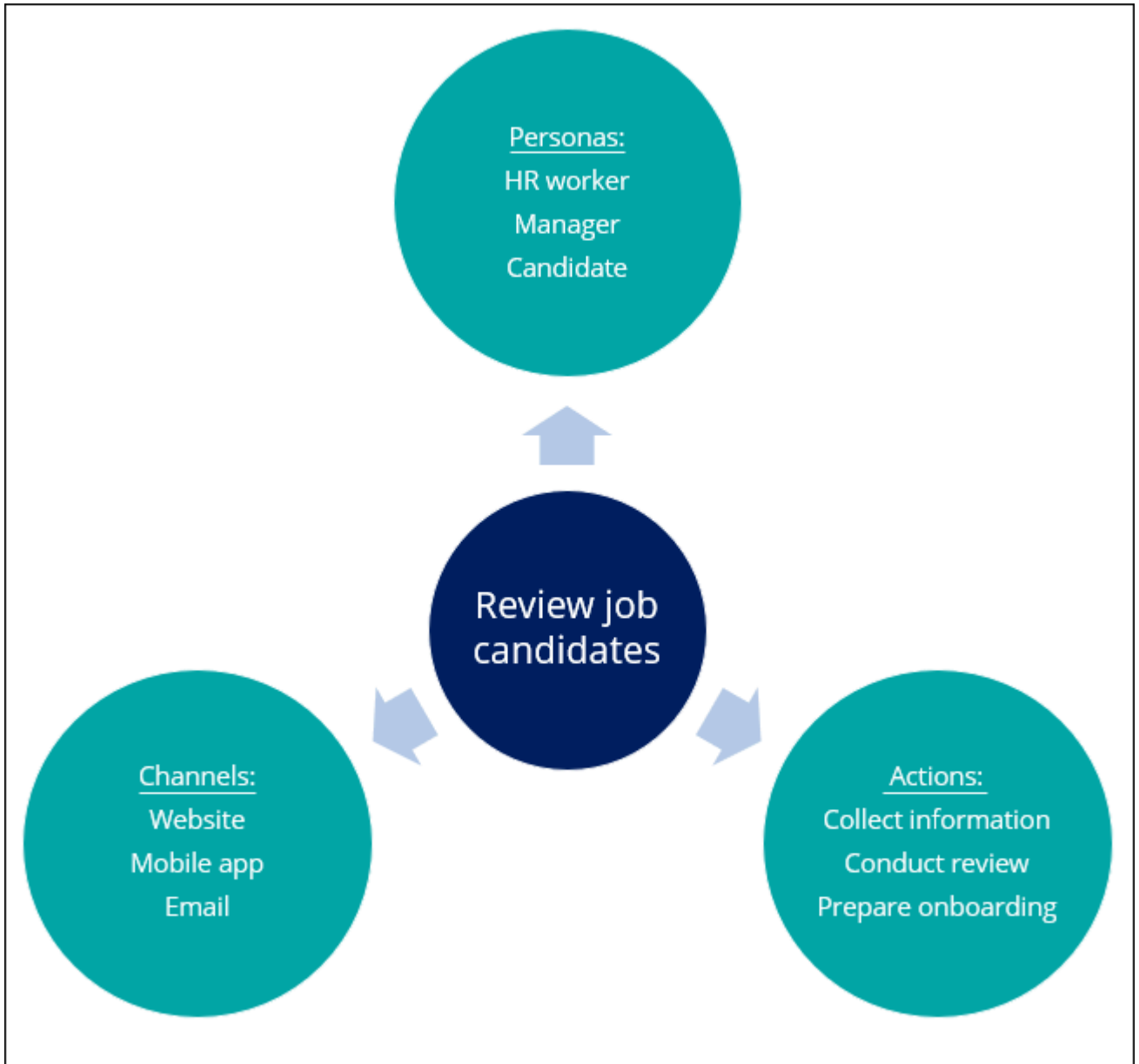
After you define the personas in your business process and what actions they need to perform, think about the device that is the most efficient and relevant solution in various scenarios. Consider whether users will need to interact with your application through a website, a mobile app, or by email. For example, customers can access an application for filling in loan requests through a browser-based web channel, or through a mobile channel that relies on a mobile app. In addition, you can also build a Facebook chatbot for customers who want to ask questions about their requests. At the same time, you can set up a second, more utilitarian web channel to accommodate professional users, such as CSRs. By making the application accessible over several channels, you can reach a wide variety of audiences, and provide each user group with the most appropriate content and user experience.

The following figure shows a diagram that includes basic elements of an application to review job candidates. The diagram includes personas involved in the



process, main actions in the process, and interactive channels that personas can

use:



*Main elements of a sample application*

**What to do next:** After you define the crucial elements of your business processes, start modeling and visualizing your work. See [Building case types \(on page 20\)](#).

### Related information

[Configuring case participants \(on page 240\)](#)

[Notifying participants about events \(on page 235\)](#)

[Mobile features overview \(on page \)](#)

[Enabling case processing in offline-enabled mobile apps \(on page \)](#)

[Conversational channels \(on page \)](#)

## Building case types

After you identify the main elements of your case types, and the relationships and dependencies in a case type hierarchy, start dividing your business cases into smaller parts so that you can use Pega Platform to visualize and then process your work to achieve your goals.

In Pega Platform, when you model a path that your case follows to a resolution, you define a case life cycle that consists of tasks grouped in a logical and practical way. By taking this approach, you can design your application to function in the same way that you think about your work. Pega Platform offers you flexibility to seamlessly move between tasks in an order that matches your unique business requirements, so that you can efficiently react in changeable situations.

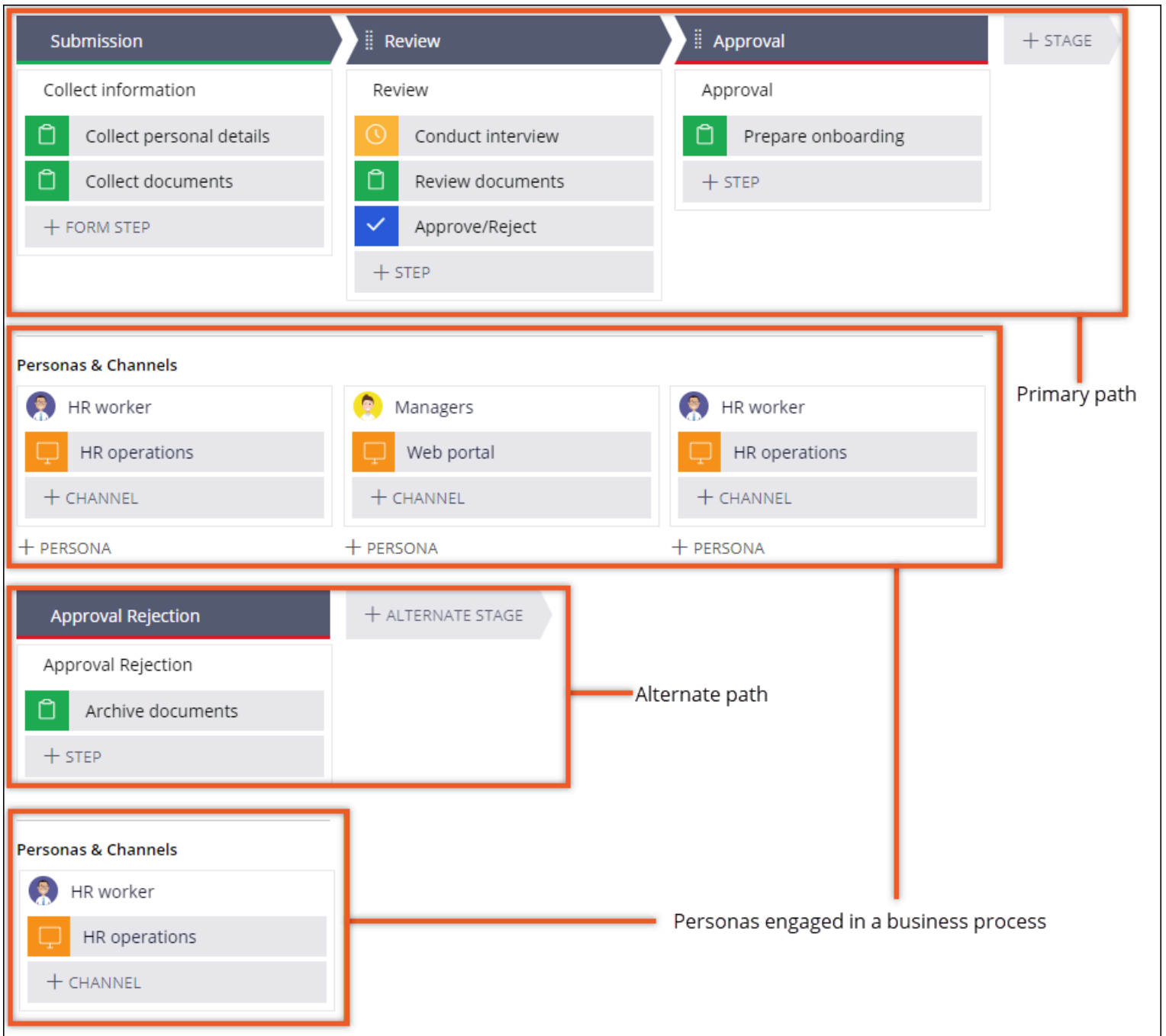
When you model your business process, divide work into the following elements:

## Stages

A stage is the first level of organization for the tasks that are required to complete a work process. Stages visualize milestones or significant events in a case life cycle. Stages can also indicate a transition of work from one person to another. For example, in a case type for Job application reviews, you can create the following stages:

- Submission, in which an HR worker collects personal details and relevant documents from a job applicant.
- Review, in which a hiring manager conducts an interview with the job applicant and reviews submitted documents.
- Approval, in which an HR worker prepares onboarding information for the approved new hire.

The stages that are necessary to resolve a case by following a default path are primary stages. To increase the flexibility of your application and resolve cases that alter from the default path, you can add alternate stages to your case life cycle. For example, in the job application review process, you can create a Rejection alternate stage that the case follows after a hiring manager rejects a candidate, as in the following example:



Case Designer displaying primary stages and an alternate stage

Modularity in case types provides for more flexible work processing, because a case can reenter any stage, for example for additional input, or enter a stage only when the stage is relevant to a specific scenario.

To define a stage, which represents a milestone in your business case, you create processes that are collections or individual tasks.



## Processes

A process consists of a series of tasks, or steps, and visualizes a set of actions within a stage. You can create a sequential process that is a basic set of tasks, and for each stage you can create multiple processes. By creating multiple processes, you group tasks into logical phases, instead of having a list of tasks that might seem to be loosely connected. For example, for the Submission stage of a recruitment process, you can add processes for Collect personal information and Collect documents. By adding multiple processes, you also model the order in which case workers complete tasks, because a case moves to the next process when all of the steps from the previous process are complete. To speed up case processing, you can also create parallel processes that can involve more case workers simultaneously. For example, one HR worker might collect personal details from a job applicant, while another HR worker collects the applicant's documents at the same time, as in the following example:



Parallel processes in Case Designer

Creating processes saves time because you can reuse a process in different stages and case types.

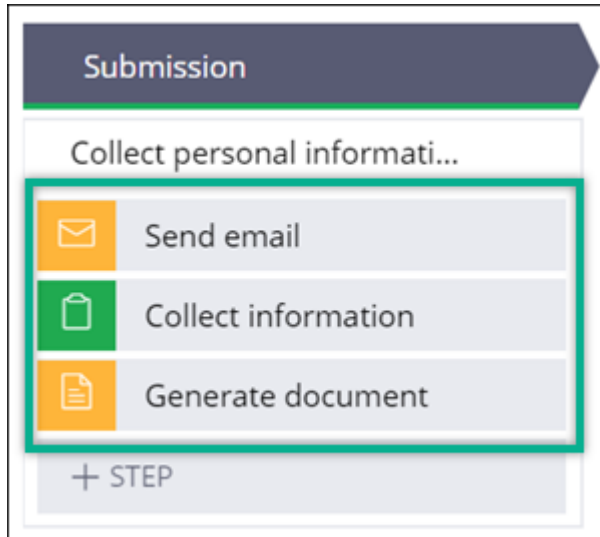
You populate processes by adding steps.

## Steps

Steps are the smallest elements of a case life cycle and represent single tasks or assignments. A step can be a user action or an automation that an application performs. Pega Platform offers a wide choice of both user actions and automations that you can add to your case life cycle. As a result, you can model case types that exactly meet your business requirements. For example, for a Collect personal information process, you can add the following steps:

- Send email, which is an automation that sends a message to a job applicant.
- Collect information, which is a user action that a job applicant performs by providing personal details.
- Generate document, which is an automation that creates a document with an applicant's details.

The following figure shows visual representation of tasks in a process:



Steps in a process in Case Designer

You can modify additional options for steps, such as the content of an email that the application sends.

## Reusing assets in case types

To save time when you create a case type, you can reuse assets from an existing case type immediately upon creation. You can reuse data, views, and even an entire life cycle. For greater flexibility, you can go further and modify reused elements. For example, after you create a case type for reviewing job candidates, you can reuse its case life cycle in a case type for reviewing candidates for managerial positions, and then modify it to add additional actions.

Ensure that users have enough information to successfully resolve cases. For more information, see [Providing data for cases \(on page 24\)](#).

For relevant training materials, see the [Defining a customer Microjourney](#) module on Pega Academy.

### Related information

[Case life cycle elements \(on page 43\)](#)

[Stages in a case life cycle \(on page 46\)](#)

[Processes in a case life cycle \(on page 66\)](#)

[Steps in a case life cycle \(on page 78\)](#)

## Providing data for cases

After you define templates for your business processes, in order to start processing work you need data. Pega Platform offers tools that you can use to efficiently collect, organize, and manage data in your application.

For data operations, case management relies on data integration. When you want to provide data for your cases, consider the following best practices:

## Collecting information from users

One method of gathering data is collecting information from users. In Pega Platform, you can create forms that users complete to provide you with necessary details. To ensure that case workers have all relevant information, define what fields you need to include in a form and what data format you need. For example, in a loan request case, a form might include fields to provide a name, surname, date of birth, address, a phone number, and a loan amount. Additionally, you can specify that certain fields consist of only certain characters. For example, a phone number field that consists of digits only. Each field represents a data entity that your application pairs with a value that a user provides. Pega Platform refers to data entities as data objects.

## Reusing data across your application

To save time and maintain consistency, you can reuse data objects and values that you need to reference across your application. For example, when a user provides a name and a surname, you can include those entities in any further context that you need, such as an autogenerated email or case documents. For improved data management, you can organize data objects into groups, and then reuse the groups across your application. For example, you can create a Loan Request Applicant data object that includes the applicant's personal details, such as a name, a surname, and contact information, as well as their income details. Every time that you use the Loan Request Applicant data object, you then ensure that you provide an entire set of values.

The following figure shows sample scenarios in which you can reuse personal details of a customer across your application. For example, you can reuse the details on



certain forms or for cases related to filing a complaint or returning purchased objects:



For greater efficiency and flexibility, you can reuse data across multiple applications and systems. Consider a scenario in which a company uses different applications for handling timesheets, expense reports, and facility requests. Instead of creating duplicate data objects, the applications can fetch data from one database – a process that is possible because of data virtualization. By virtualizing data, you not only lower development time and costs, but also provide meaningful and consolidated data in multiple scenarios.

## Integrating with external systems

To save time and reduce development costs, you can also create integrations between your application and external storage systems. Consider a scenario in which a financial services provider handles the payroll for employees of multiple companies. Instead of manually providing data about each employee, you can create a data object that uses an integration, such as a REST API, to source the required information from an external system of records. For example, from a database that stores information about employees of a particular company. Because of the integration, the financial services provider can process their work without having direct access to the external database. When an employee changes their details in the external storage system, you can ensure that your application reflects the changes. As a result, you increase data consistency and avoid errors.

**What to do next:** Ensure that your business processes align with regulations and policies within your company. See [Understanding and complying with case management best practices \(on page 26\)](#).

### **Understanding and complying with case management best practices**

As you work through your business process, you might need specific information or data formats to reach your business goal. Pega Platform offers tools that you can use to ensure that the work that users perform in your application aligns with compliance and regulatory requirements that you set for your case types.

You can apply the following best practices to ensure that your case types follow rules and policies that you define for your work:

## **Building a case life cycle**

To provide users with a series of logically ordered steps towards their business goal, define a case life cycle that orders actions into a sequence. For example, in an application for reviewing job candidates, you can define a sequence in which a hiring manager needs to approve a candidate before an HR worker starts an onboarding process for the candidate. Although with Pega Platform you can create flexible applications and respond to dynamically changing circumstances in an agile way, you can also define which actions are necessary to resolve a business process. As a result, you ensure that case workers have enough data to resolve a case in any scenario, and that your work meets all the regulations specific to your organization.

For more information, see [Case life cycle elements \(on page 43\)](#).

## **Eliminating duplicate cases**

To save time and maintain the good quality of processing, you can search for and then eliminate duplicate cases. As a result, you ensure that every case is unique, and you increase the chances of reaching a successful resolution. You can define conditions that immediately evaluate whether a new case is a duplicate. For example, you can ensure that a customer can create only one loan request for a certain amount of money before resolving a previous request. In this way, you avoid unnecessary processing and maximize the probability of meeting your customer's business objectives.

For more information, see [Searching duplicate cases \(on page 109\)](#).

## **Validating case data**

To avoid errors and speed up case resolution, assist users when they provide data for cases. For example, when a user enters a phone number, an application checks whether the number includes the correct number of digits and no special characters, so that a customer service representative (CSR) can contact the customer when required. You can create validation conditions that meet your unique business requirements and provide error messages so that users can quickly locate the problem and provide their data in the correct format.



For more information, see [Validating case data \(on page 164\)](#) and [Validating field values on a form \(on page 183\)](#).

## Auditing case data

To maintain case compliance and follow progress, you can conveniently track changes that users make in cases. For example, you can check how and when a user changes a shipping address in a purchase order case, so that their parcel of purchased items reaches the correct place. You can track changes to specific fields, as well as view audit messages that users submit upon performing a change.

For more information, see [Enabling field-level auditing \(on page 212\)](#).

## Archiving case data

To save on your database costs and comply with regulatory requirements at the same time, create an archival policy for your cases. You can archive inactive cases and remove them from the database to a secondary storage. For any audit purposes, you can still search for and view the inactive cases. For example, you can archive any case that was resolved more than a year ago. If you need to audit the case, for example because a customer files a complaint, you can automatically access the case from your secondary storage.

## Providing case access

To ensure that users interact with relevant case data only, you can configure case access settings. By exposing only applicable actions and information to users, you deliver an application that is flexible and easy to follow at the same time. You also protect confidential data from random users and ensure that only appropriate users can perform certain actions. For example, to review loan requests in an application, only managers might have an option to approve or reject a request. In a different scenario, sales representatives can access the details of only their own sales opportunities.

For more information, see [Securing case access \(on page 453\)](#).

---

### Related information

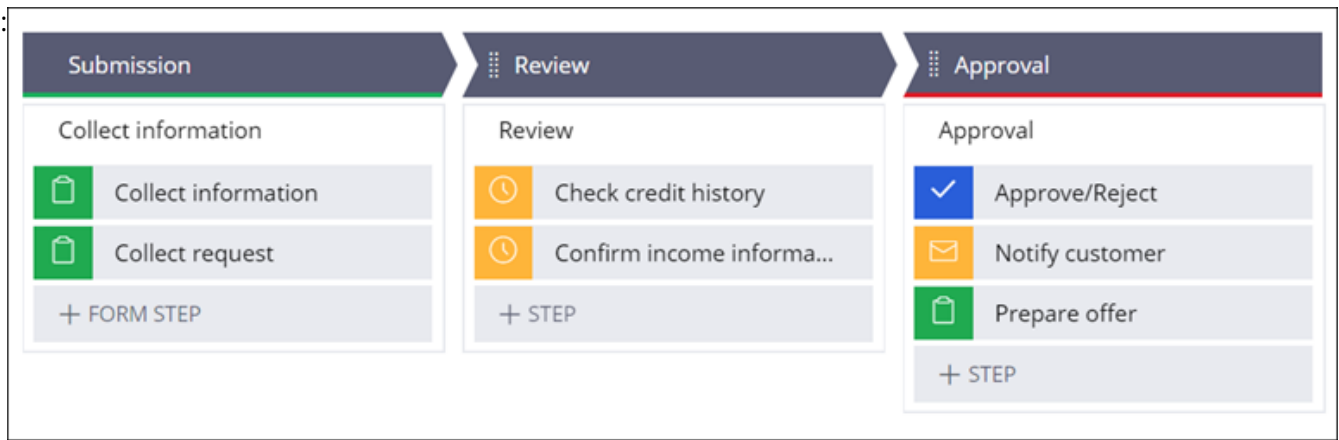
[Configuring case type settings \(on page 203\)](#)

## Automating work by creating case types

Complete your Microjourney and meet your business goals in a convenient, time-saving, and automatic way by creating case types. By creating case types, you define how users of your application manage and resolve work to achieve a successful outcome in the most efficient way.



When you create a case type, you create a visual representation of your Microjourney that consists of related tasks that are grouped into coherent stages. Each stage consists of steps that users of your application, such as customer service representatives (CSRs), need to complete to bring a case closer to its resolution. To save time, you can also define automated steps that your application performs, and then set goals and deadlines for these actions. Cases are instances of case types, which are reusable templates of your business processes. For example, you can create a case type that CSRs at a bank might reuse every time they process a loan request. An individual loan request case might then include stages such as collecting information from the customer, checking their credit history, and approving or rejecting the loan request. As a part of this case life cycle, an application can autogenerate an email to the customer with the bank's decision about their loan, as in the following example:



A sample case life cycle in Case Designer

To further maximize the efficiency of your business actions, you can define case types to meet your specific needs, by specifying process behavior throughout different stages of the case life cycle. You can configure how users interact with a case and in what manner the case's automated processes facilitate this interaction, helping your customers more quickly reach their business goals.

For relevant training materials, see the [Defining a customer Microjourney](#) module on Pega Academy.

---

Creating unit test cases for flows and case types (on page )

## Case types

Save time and maximize efficiency when you implement similar business processes by defining case types. A case type is a visual representation of your business process, and a template for work that you can reuse for multiple instances of your business process. By creating a case type, you define the ultimate goal of a business process, the path that the case must follow to a resolution, the people who are involved in processing the case, and the data that the case requires. For example, in an accident claim, you define actions such as collecting documents and reviewing vehicle damage, people such as a policyholder and an insurance company representative, and data such as a vehicle identification number and vehicle registration plate. When you create reusable work templates, you save time and ensure that your business processes reach successful resolution.

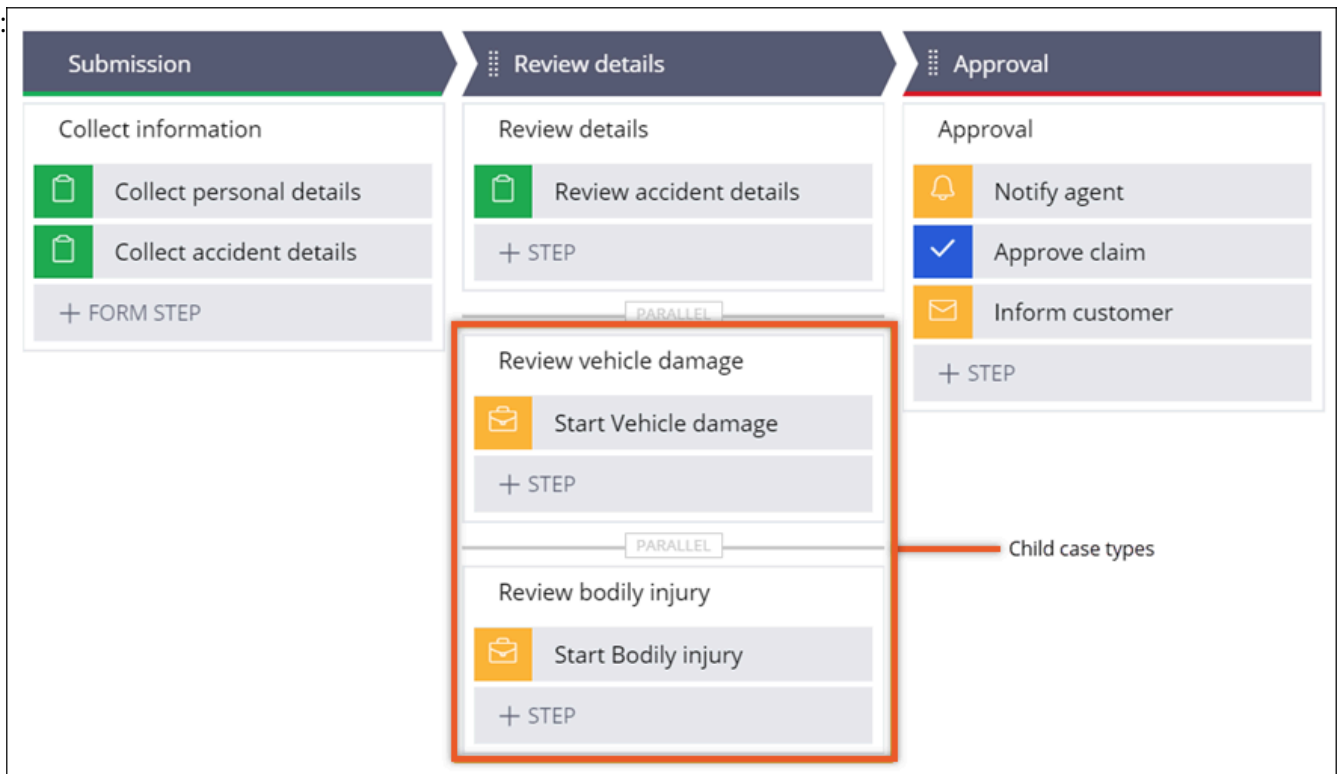


Pega Platform refers to a main business process as a parent case type, and to supporting business processes as child case types. A parent case type depends upon the results of the child case types that are together needed to reach the ultimate, expected outcome. To resolve a parent case, you first need to resolve all of its child cases. For example, a parent case type that represents an Accident claim can only be resolved after child case types for Vehicle damage and Bodily injury are completed.

In this example, an insurance company representative creates an Accident claim case (the parent case) each time a policyholder reports a car accident. After your application verifies the policyholder's driving license and vehicle identification number, the system creates a Vehicle damage case (the child case) that supports the Accident claim case.

The insurance company representative then decides whether the application also needs a Bodily injury case, based on the information from the policyholder. After the claims adjusters process the Vehicle damage and Bodily injury cases, the system automatically calculates the funds payable and updates the policy premium in the Accident claim case. Agents who can approve the claim then receive a notification that the Accident claim is ready for review.

The following figure shows a sample business process that includes two child case types:



A parent case type that includes two child case types in Case Designer

Save time, reduce costs and increase the flexibility of your application by reusing child case types in different scenarios. For example, both Vehicle and Property insurance claims can include a Bodily injury claim. By implementing the Bodily injury claim as a separate case type, you can use the Bodily injury case type with both the Vehicle and Property insurance claims.

## Naming conventions for case types

When you select a name for your case type, focus on the outcome of your business process rather than on the actions that users need to complete. The structure of case types is flexible and can adjust to changing business circumstances, when the goal remains unchanged. Ensure that you provide names that clearly convey the purpose of your case type. For example, the name Financial operations is ambiguous and non-specific, but Loan application review, Mortgage request review, or Credit card dispute precisely communicate their purpose.

Creating a Microjourney for customer success (*on page* )

[Deleting case types \(on page 317\)](#)

[Renaming case types \(on page 316\)](#)

Automatically included case types (*on page* )

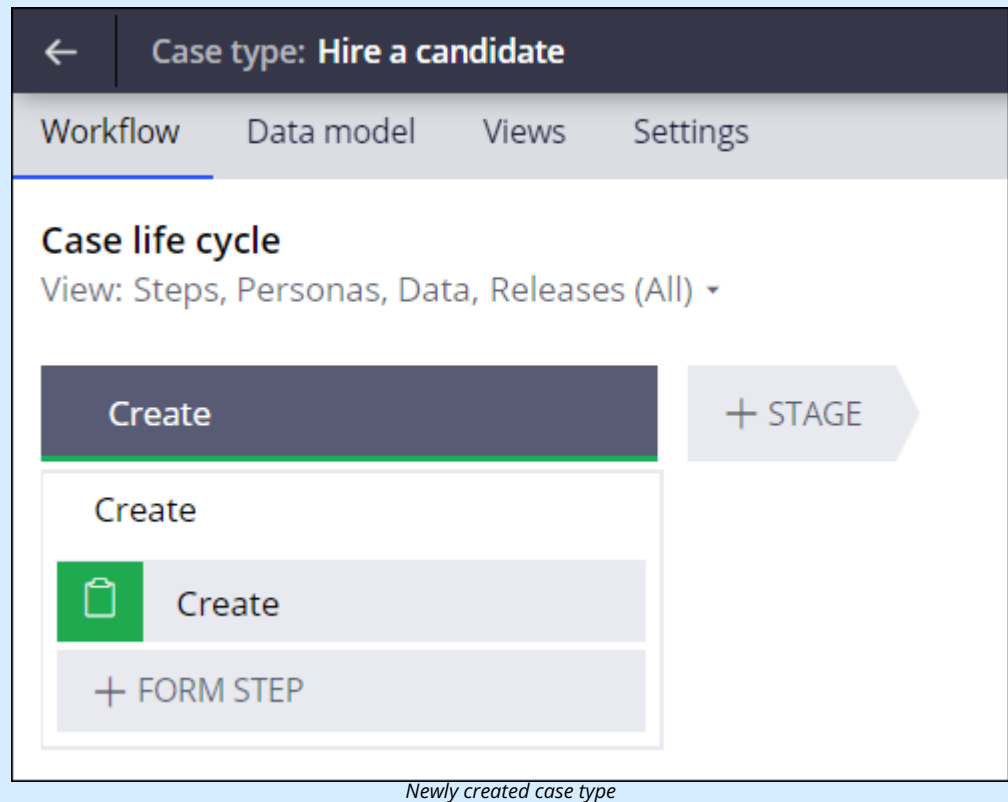
### Creating a top-level case type

Improve work processing in your application by creating top-level case types that visualize business processes. When you visualize a business process, you can conveniently divide the process into phases, and then track and manage work with greater ease. For example, you can create a top-level case type that represents a hiring process, and then reuse the type every time when you want to process a job application from a new candidate.

1. In the navigation pane of Dev Studio, click **Case types**.
2. In the navigation pane of App Studio, click **Case types**.
3. Click **Add a case type**.
4. In the header of the **Case types** section, click **New**.
5. In the **Name** field, enter a name for the case type.
6. In the **Case type name** field, enter a name for the case type.
7. In the **Type** list, select **Standard**.
8. **Optional:** To reuse assets from another case type, click **Advanced**, and then in the **Reuse assets from** list, select a case type.
9. **Optional:** To change the inheritance model for this case type, expand the **Advanced Settings** section, update the **Derives from (Directed)** and **Derives from (Pattern)** inheritance settings, and then provide rule sets and rule set versions.  
The inheritance model defines classes from which your case type can inherit data. For more information, see [Understanding class hierarchy and inheritance \(on page \)](#).
10. Click **Submit**.
11. Click **Next**.



Your new case type includes a default Create stage with a Create process, as shown in the first figure. The second figure shows the same case type already populated with stages and processes.



[The Create stage \(on page 48\)](#)

[Case types \(on page 28\)](#)

[Case life cycle elements \(on page 43\)](#)

[Creating a data model from a spreadsheet \(on page \)](#)

[Creating a Microjourney for customer success \(on page \)](#)

## Creating a child case type

Create child case types to ensure that users of your application address all elements in a complex business process before resolving the case. When you supplement your business processes with child case types, you create dependencies that these case types represent. For example, you can create a case type that represents a review of an accident claim, and then supplement the case type with child case types that visualize the processes of estimating vehicle damage and injuries. To resolve the `Accident claim` case, a worker first resolves the `Vehicle damage` and `Injury` cases. Different workers can handle each child case in parallel to resolve cases more efficiently.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the process in which you want to add a child case type, click **Add step to the process > More > Automations > Create case**, and then click **Select**.
3. In the **Create case** field, enter the name of the child case type, for example `Vehicle damage`.
4. In the **Step** navigation pane, in the **Create the following case** list, select a type of a child case type that you want to create:
  - To create a new child case type, select **Create new case type**, enter a unique name for the case type, and then click **Create**.
  - To reuse an existing case type, select a case type.
5. **Optional:** To create more child case types, select **Create multiple cases using a list**, and then, in the **List field** list, select a value that determines the number of cases to create.
6. **Optional:** To reuse information from another case, select the **Transfer information to new case** check box, and then, in the **Transfer information** window, select the information that you want to transfer:
  - a. In the **'From' field** column, select the field that stores the values that you want to transfer.
  - b. In the **'To' field** column, press the Down arrow key, and then select the destination field to which to transfer the value.
  - c. **Optional:** To add mapped fields to the case type view, in the **View** section, select the **Add mapped fields to [case type view name] view** check box.
  - d. Click **OK**.
7. Click **Save**.

At run time, the fields that you select as target fields in a child case type, consist of information from the parent case type.

[Case types \(on page 28\)](#)

[Understanding case hierarchy \(on page 14\)](#)

[Building case types \(on page 20\)](#)

[Creating contextual cases \(on page 106\)](#)

## Creating child case types in Dev Studio

Define a more structured work processing in your organization by creating child case types. When you create a child case type, you define a set of actions that are dependent on the main business process, and that users need to complete before the parent case type reaches a resolution. As a result, you avoid creating long and complex case types and you deliver more granular and flexible applications that are easier to maintain. For example, a process of hiring an employee for a managerial position might require more complex and longer background check. By defining the background check as a child case type, you





have the possibility to reuse this business scenario in only the relevant circumstances. A case of hiring a manager can reach a resolution only after the child case of running a background check completes.

**Note:** Creating case type hierarchy in Dev Studio is suitable for advanced developers. To take advantage of low-code tools, create child case types in App Studio. For more information, see [Creating a child case type \(on page 31\)](#).

1. In the navigation pane of Dev Studio, click **Case types**.
2. Hover over the name of a case type, and then click **More > Add a child case type**.
3. In the **Add case type** dialog box, define the case type that you want to use:
  - To create a new case type, select **New case type**, and then, in the **Name** field, enter a label for your case type.
  - To reuse an existing case type, select **Existing case type**, and then, in the **Select an existing case type to add as a child** list, select a case type that you want to use.
4. **Optional:** If you create a new case type, to define resources that your case type can access by using class inheritance, expand the **Advanced Settings** section:
  - To change the parent class from which your case type directly inherits resources, in the **Derives from (Directed)** field, enter a new class.
  - To change the class from which your case type inherits resources by using the class name structure, in the **Derives from (Pattern)** list, select a new class.
  - To change the ruleset or the ruleset version to store your new case type, in the **Ruleset** and **Ruleset version** lists, select the new values.
5. Click **Submit**.

[Understanding case hierarchy \(on page 14\)](#)

[Identifying case types elements \(on page 9\)](#)

[Case types \(on page 28\)](#)

[Understanding class hierarchy and inheritance \(on page \)](#)

## Creating a case type rule form in Dev Studio

Provide advanced options for your business processes by creating case types through a case type rule form. For example, you can select an application layer in which to store your case type, or define multiple options at once, for example, for the people that your case type involves, processes that can support your case type, or dependencies with other case types.

Creating case types through a rule form is suitable for advanced developers. If your use case does not require advanced configurations, create a case type by using the Case Designer tool, which helps you



visualize your business process in a low-code way. For more information, see [Creating a top-level case type \(on page 30\)](#).

1. In the header of Dev Studio, click **Create > Process > Case Type**.
2. In the **Case Type Record Configuration** section, in the **Label** field, briefly describe the purpose of your case type.
3. In the **Context** section, define how your application stores the case type:
  - a. If you use branches to work on your application, in the **Development branch** list, select a branch that you want to use to store the case type.
  - b. In the list of application layers, select a layer in which you want to store the case type.
  - c. In the **Applies to** field, press the Down arrow key, and then select the class to store the case type.
  - d. In the **Add to ruleset** list, select a ruleset and a ruleset version to store the case type.
4. **Optional:** To override the default work item that your application associates with this development change, press the Down arrow key in the **Work item to associate** field, and then select a work item.
5. Click **Create and open**.
6. **Optional:** To customize how the case appears at run time, change the icon that the system displays to users when processing the case:
  - a. In the **Appearance** section, click **Edit**.
  - b. In the **Appearance** dialog box, select a new icon.
  - c. Click **Submit**.
7. **Optional:** To define people involved in a case by reusing an existing rule, in the **Work parties rule** field, provide a relevant rule name.  
You can also define work parties manually. For more information, see [Configuring case participants \(on page 240\)](#).
8. In the **Starting process** section, click **Add a row**, in the text field press the Down arrow key, and then select a process that runs when the case starts at run time.  
You only provide a starting process for case types without the Create stage. For more information, see [The Create stage \(on page 48\)](#).
9. **Optional:** To enable users to supplement the case with additional processing, in the **Case wide supporting processes** section, in the text field press the Down arrow key, and then select a supporting process that users can add to this case.
  - a. **Optional:** To add more supporting processes, click the **Add a row** icon, and then provide another process.
10. **Optional:** To create dependencies between case types, in the **Child case types** section, enter a case type that users need to resolve before the current case reaches its resolution.  
For more information about configuring child case types on a rule form, see [Creating case hierarchy on a case type rule form \(on page 35\)](#).
11. In the **Case wide actions** section, define how users interact with a case by providing a flow action:

- a. In the **Flow Action** column, enter a flow action.
  - b. In the **Visibility** column, define when a flow actions starts by selecting a When rule, an expression, or an option to always start the flow action.
12. **Optional:** To look for similar cases in the database so that you avoid creating duplicate cases, in the **Case match** section, in the text field, enter a rule name that evaluates conditions used for finding duplicate cases.
  13. Click **Save**.

---

[Case types \(on page 28\)](#)

[Introduction to case management \(on page 6\)](#)

[Adding supporting processes to cases \(on page 357\)](#)

[About Flow Actions \(on page \)](#)

[Searching duplicate cases \(on page 109\)](#)

## Creating case hierarchy on a case type rule form

Organize hierarchically your business processes by adding child case types to your main, parent case type. Consequently, you define which additional actions users need to resolve before your main process is complete. For greater flexibility and efficiency, you can define conditions under which a child case type starts. When you use a case type rule form to create case hierarchy, you can select advanced options and perform additional configuration. For example, you can supplement your parent case type *Review a job candidate* with a child case type *Run a background check* to provide more information about a candidate who applies for a position in your organization. To make the process more flexible, you can run a background check on some of the candidates, for example, when you want to hire a manager.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Case Type**.
3. In the list of case types instances, open the case type that you want to edit.
4. On the **Process** tab, in the **Child case types** section, in the text field, press the Down arrow key, and then select a case type that you want to use in the case type hierarchy.
5. **Optional:** To define how case types in the hierarchy exchange data, click **Data propagation**, and then define the sharing options for the data:
  - a. In the **Data Propagation** dialog box, click **Add property values**.
  - b. In the **Propagate property value** column, enter a source property from the parent case type.
  - c. In the **To property value** column, enter a destination property in the child case type.

At run time, your application populates the contact number in a `Run` background check child case type using a value that a user provides for a phone number in a `Review job candidate parent` case type.

- d. **Optional:** To propagate values for more properties, repeat steps [5.a \(on page 35\)](#) through [5.c \(on page 35\)](#).
  - e. **Optional:** To use a data transform to pass the values, select the **Apply data transform** check box, and then, in the text field, enter a data transform.
  - f. Click **Submit**.
6. **Optional:** To define when the child case starts, click the **Disclose** icon, and then define case instantiation options:

Choices	Actions
<p><b>Allow users to start a child case manually at run time</b></p>	<ol style="list-style-type: none"> <li>a. Select the <b>Manual instantiation</b> check box.</li> <li>b. <b>Optional:</b> To allow users to start the case only under certain circumstances, in the <b>Permitted When</b> field, enter a When rule that defines when a user can start a child case.</li> </ol>
<p><b>Start a child case automatically</b></p>	<ol style="list-style-type: none"> <li>a. Select the <b>Automatic instantiation</b> check box.</li> <li>b. Select whether a child case starts together with a parent case or if a child case starts when a case fulfills the dependencies.</li> <li>c. <b>Optional:</b> If you configure a case type to start during a parent case creation, to allow case instantiation only under certain circumstances, in the <b>Permitted When</b> field, enter a When rule that defines when the child case starts.</li> </ol>

- Tip:** You can select both manual and automatic start of a case type. If the system cannot
- ① start a case type automatically, a user can start the case manually. If you skip defining case instantiation options, the system always starts the child case.

7. **Optional:** To define additional child case type settings, click the **Disclose** icon, and then provide relevant values:
  - a. To define how many instances of a child case the application can start at run time, in the **Max Instances** field, enter an integer.
  - b. **Optional:** To indicate that the parent case type cannot reach resolution before the child case type is complete, select the **Required** check box.
8. **Optional:** To add more child case types, click **Add a row**, and then repeat steps 4 (on page 35) through 7 (on page 37).
9. Click **Save**.

### Related information

[Understanding case hierarchy \(on page 14\)](#)

[Case types \(on page 28\)](#)

[Creating a child case type \(on page 31\)](#)

## Calculating properties in a case type

Save time and manage your case type resources efficiently by automatically calculating aggregated properties. You can calculate a property by sourcing values from other case types instead of asking users to provide values manually. For example, in an `Insurance claim` case type, you can calculate the total insurance amount by adding the values for the insurance amount from `Vehicle damage` and `Bodily injury` child case types.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Case Type**.
3. In the list of case type instances, select the case type that you want to edit.
4. On the **Calculations** tab, in the **Aggregate properties**, expand the **Property to calculate** section.
5. In the **Calculated property** field, enter a destination property that is a sum of other properties. Select a single value property of an integer or decimal type.
6. In the **Case type** column, select a case type that stores a source property.
7. In the **Property** column, enter the source property that you want to use to provide the value.

8. **Optional:** To add more properties, click **Add property**, and then repeat steps [6 \(on page 37\)](#) through [7 \(on page 37\)](#).
9. **Optional:** To add more aggregated properties, click **Add calculation**, and then repeat steps [5 \(on page 37\)](#) through [7 \(on page 37\)](#).
10. Click **Save**.

[Validating case data \(on page 164\)](#)

[Referencing properties \(on page \)](#)

[Building logic and calculating values in your application \(on page \)](#)

## Configuring stages on a case type rule form

Provide advanced configurations for sets of actions in your business processes by adding stages on a case type rule form. As a result, you can configure multiple settings from a single screen. For example, you can define which optional processes you want to enable for the case type, or what attachments users need to provide in order for the case to enter a specific stage. In a sample scenario with a `Review insurance claims` case type, you can add primary stages that represent collecting information, reviewing gathered data, and making a decision about rejecting or approving a claim. Primary stages represent an expected path of a business process. You can also add an alternate path to define actions that happen in a case when an exception occurs. In the `Review insurance claims` case type, for example, you can create an alternate stage that the case enters after rejection of the claim.

Adding stages through a rule form is suitable for advanced developers. If your use case does not require advanced configurations, populate your case types by using the Case Designer tool, which helps you visualize your business processes in a low-code way. For more information, see [Stages in a case life cycle \(on page 46\)](#).

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Case Type**.
3. In the list of case type instances, select the case type that you want to edit.
4. On the **Stages** tab, add a stage to your case type:
  - To add a primary stage, in the **Primary Stages** section, click **add primary stage**, and then in the text field, enter a stage name.
  - To add an alternate stage, in the **Alternate Stages** section, click **add alternate stage**, and then in new text field, enter a stage name.
5. Configure conditions for skipping a stage:

- To ensure that a case always enters a stage, in the **Skip stage** list, select **Never**.
- To skip a stage by evaluating a When rule, in the **Skip stage** list, select **When Rule**, and then in the text field, enter a When rule name.
- To skip a stage by using an expression, in the **Skip stage** list, select **Expression**, and then in the text field, enter an expression name.




**Note:** You can configure skip conditions for primary stages only. A case cannot skip the last stage.

#### 6. Configure case behavior when a stage is complete:

Choices	Actions
<p><b>Move the case to the next stage</b></p>	<p>In the <b>When all processes in this stage are complete</b> section, select <b>Automatically move to next stage</b>.</p> <p><b>Note:</b> This option is available for primary stages only.</p>
<p><b>Allow users to perform optional actions before moving the case to the next stage</b></p>	<p>In the <b>When all processes in this stage are complete</b> section, select <b>Wait for a user action</b>.</p>
<p><b>Resolve the case</b></p>	<ol style="list-style-type: none"> <li>In the <b>When all processes in this stage are complete</b> section, select <b>Resolve the case</b>.</li> <li>In the <b>Resolution status</b> field, enter a status that a case enters upon resolution.</li> <li><b>Optional:</b> To delete any work that remains after the case resolution, select the <b>Delete open assignments</b> check box.</li> </ol>

Choices	Actions
	<p>d. <b>Optional:</b> To delete any dependent cases after the case reaches resolution, select the <b>Delete open assignments</b> check box.</p>

7. **Optional:** To display a case progress status, in the **Set case status on stage entry** field, define the status that a case acquires when entering the next stage.

 **Tip:** You can reuse existing statuses or enter a new value.

8. **Optional:** To provide enough information to process a case, in the **Requirements** section, define the requirements that the case needs to meet before entering the next stage:
- To validate data, in the **Validation** field, enter validation that the case needs to pass before entering the next stage.
  - To assign time frames to the stage and ensure that users resolve work in a timely manner, in the **Service level** field, enter a service-agreement rule.
  - To ensure that the case has at least one attachment before entering this stage, in the **Attachments required for stage entry** section, click **Add required attachment**, and then select an attachment category.
9. In the **Automatically launched processes** section, define processes that occur in the case without additional triggers:
- a. Click **Add process**.
  - b. In the **Process** field, enter a process that you want to run.
  - c. In the **Start process** list, define whether a process runs always, when a When rule evaluates to true, or when an expression evaluates to true.
  - d. In the **Start Step** list, define whether the process runs after the previous step or upon a stage entry.  
The first process in a stage always runs upon a stage entry.
  - e. **Optional:** To apply time frames for users to resolve the process, in the **Service level** field, enter a name of a service-level agreement rule that you want to use.
  - f. **Optional:** To add more processes, repeat steps [9.a \(on page 40\)](#) through [9.e \(on page 40\)](#).



The Create stage has a default `CreateForm_Default` automatically launched process. For more information about the Create stage, see [The Create stage \(on page 48\)](#).

10. **Optional:** To increase flexibility and provide additional processing in a case, add an optional process:
  - a. In the **Optional Processes** section, click **Add process**.
  - b. In the **Process** field, enter a name for the process that you want to use.
  - c. In the **Visible** list, define whether a process runs always, when a When rule evaluates to true, or when an expression evaluates to true.
  - d. **Optional:** To add more processes, repeat steps [10.a \(on page 41\)](#) through [10.c \(on page 41\)](#).
11. **Optional:** To allow users to perform out-of-sequence tasks that do not influence the main events in the case, in the **Optional actions** section, add an optional action:
  - a. Click **Add action**.
  - b. In the **Local Action** field, enter an action that you want to use.
  - c. In the **Visible** list, define whether a process always runs, when a When rule evaluates to true, or when an expression evaluates to true.
  - d. **Optional:** To add more actions, repeat steps [11.a \(on page 41\)](#) through [11.c \(on page 41\)](#).
12. Click **Save**.

---

[Building case types \(on page 20\)](#)

[Creating a primary stage \(on page 57\)](#)

[Creating an alternate stage \(on page 58\)](#)

[Processes in a case life cycle \(on page 66\)](#)

[Empowering knowledge workers \(on page 348\)](#)

[When condition rules \(on page \)](#)

## Categorizing attachments on a case type rule form

Organize the information and materials in your case types by creating categories for attachments. When you use a case type rule form to create attachment categories, you can access additional data about a category, such as a property that you can later use to reference attachments by category. For example, in a `Vehicle damage insurance claim` case type, you can create an attachment category for photos of vehicle damage that a customer service representative (CSR) needs to examine when reviewing an insurance claim. You can then easily access attachments in this category by referencing a relevant property. To meet your specific business needs, you can also create multiple attachment categories for every case type.



Categorizing attachments through a rule form is an option suitable for advanced developers. You can also create attachment categories in Case Designer, in the case type settings. For more information, see [Categorizing case attachments \(on page 457\)](#).

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Case Type**.
3. On the **Attachment Categories** tab, click **Add attachment category**.
4. In the **Category** column, provide a category name:
  - To reuse an existing category, press the Down arrow key, and then select a category that you want to use.
  - To create a new category, enter a unique category name.

The system autopopulates the description and property reference of the category.

5. **Optional:** To change the category description, in the **Description** column, enter some new text.
6. **Optional:** To ensure that a case contains at least one attachment in the attachment category when a user submits the case, select the check box in the **Required for resolution** column.
7. **Optional:** To change the name of the property, in the **Property reference** column, click **Rename**, and then, in the text box, enter a new name.
8. **Optional:** To add more attachment categories, repeat steps [3 \(on page 42\)](#) through [7 \(on page 42\)](#).
9. Click **Save**.

---

[Securing case access \(on page 453\)](#)

[Controlling access to case attachments \(on page 457\)](#)

[Referencing properties \(on page \)](#)

## Defining purpose of a case type by creating specifications

Engage with your stakeholders and communicate goals that you want your case type to achieve by creating specifications. Specifications represent actions that users can perform by using your application so that all parties involved in the development process have a common understanding of the purpose of your application.

For example, for a `Hiring a job candidate` case type, you can create specifications that describe the following actions:

- HR worker can collect personal information from the candidate
- Job candidate can upload documents
- Hiring manager can accept the candidate



Associating specifications with case types is suitable for advanced developers. To describe the purpose of your application in a low-code and more user-friendly way, create features.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Case Type**.
3. In the list of case type instances, open the case type that you want to edit.
4. On the **Specifications** tab, click **Click to associate specifications**.
5. In the **Name** column, press the Down arrow key, and then select a specification that you want to associate with the case type.
6. **Optional:** To add more specifications, repeat steps 4 (on page 43) through 5 (on page 43).
7. Click **Save**.

---

Documenting your application (on page )

Creating features (on page )

## Case life cycle elements

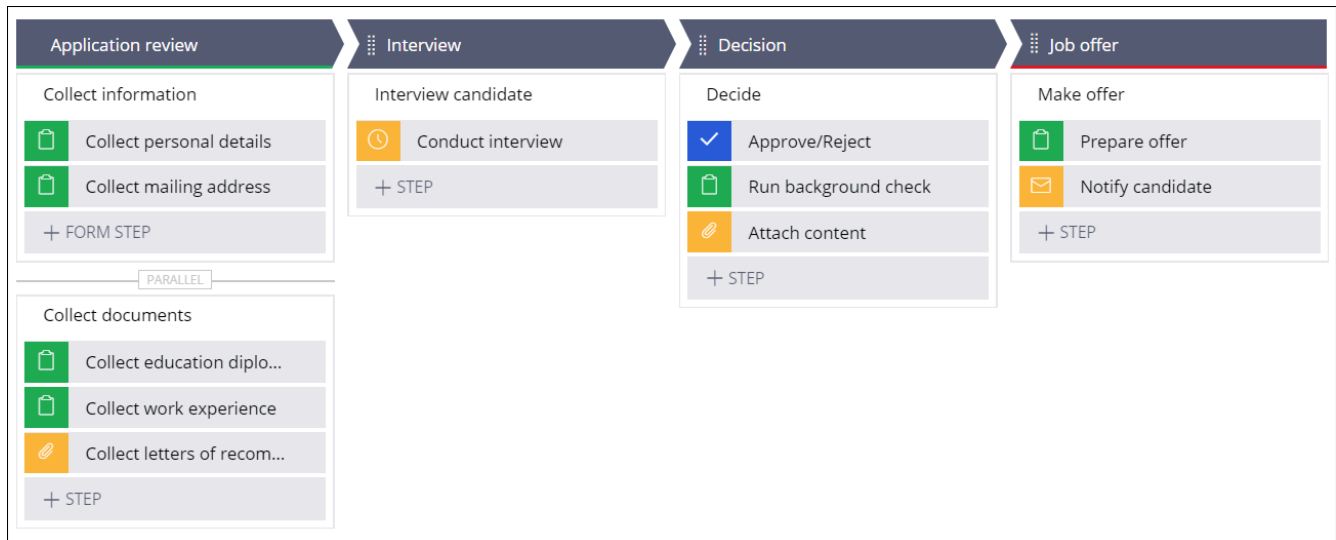
Visualize your work, organize tasks, and achieve the goals of your business process by defining the case life cycle. Creating the case life cycle is a technique that you can use to model a path that your case follows to resolution by grouping tasks in a logical and practical way.

After you identify the main elements and their relationships in your case types, start dividing your business processes into smaller parts. As a result, you can design your case type in the same way as you think about your work. Each case life cycle consists of the following elements that you can iteratively define:

- Stages that consist of processes
- Processes that consist of steps
- Steps that are either automations or user actions

## Stages

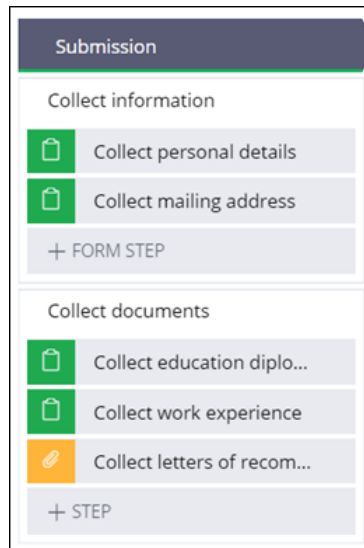
Stages are milestones in the case type, and they can indicate a transition of work from one department to another. For example, if your case type is to hire a candidate, you can create Application review, Interview, Decision, and Job offer stages, where each stage includes a set of different actions, and is performed by a different group of people, as shown in the following figure:



Stages in a case life cycle

## Processes

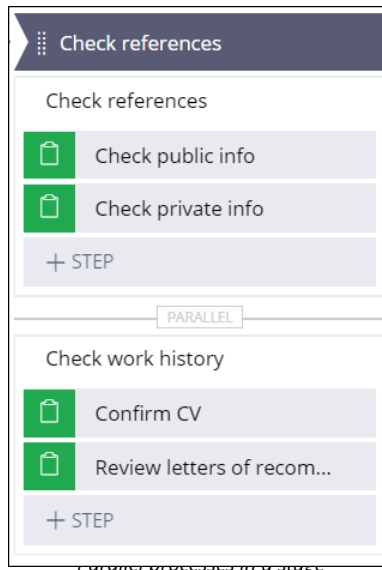
A process consists of a series of tasks, or steps, and visualizes a set of actions within a stage. For example, for Application review stage of hiring a candidate, you can add processes for Collecting personal information and Collecting documents, as shown in the following figure:



Sequential processes in a stage

By adding multiple processes, you group tasks into logical phases and model the order in which case workers complete tasks, because a case moves to the next process when all of the steps from the previous process are complete.

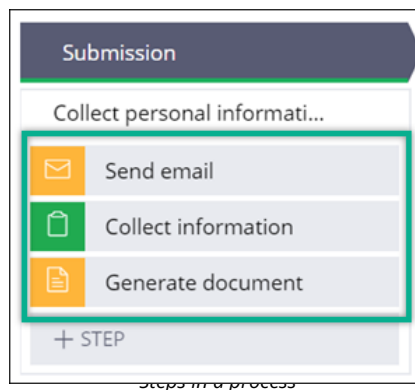
To speed up case resolution, you can configure parallel processes that involve more case workers simultaneously. For example, in the Check references stage of hiring a candidate, one HR worker conducts a reference check and the other HR worker checks work history of a candidate, as shown in the following figure:



Creating processes saves time because you can reuse a process in different stages and case types. For example, you can reuse the Collecting documents process from the reviewing job applications in other case types.

## Steps

Steps are the smallest elements of a case life cycle and represent single tasks or assignments. A step can be a user action or an automation that an application performs. For example, for a Collect personal information process, you can add a Send email step, which is an automation that sends a message to a job applicant, and a Collect information step, which is an action that a user performs, as shown in the following figure:



Define the case life cycle by completing the following actions:

---

Creating a Microjourney for customer success (on page )  
 Case types (on page 28)

## Stages in a case life cycle

Stages in your case life cycle visualize the milestones in the journey towards your business goals. By creating stages, you organize work into sequential and logical phases that help you achieve your business objectives.

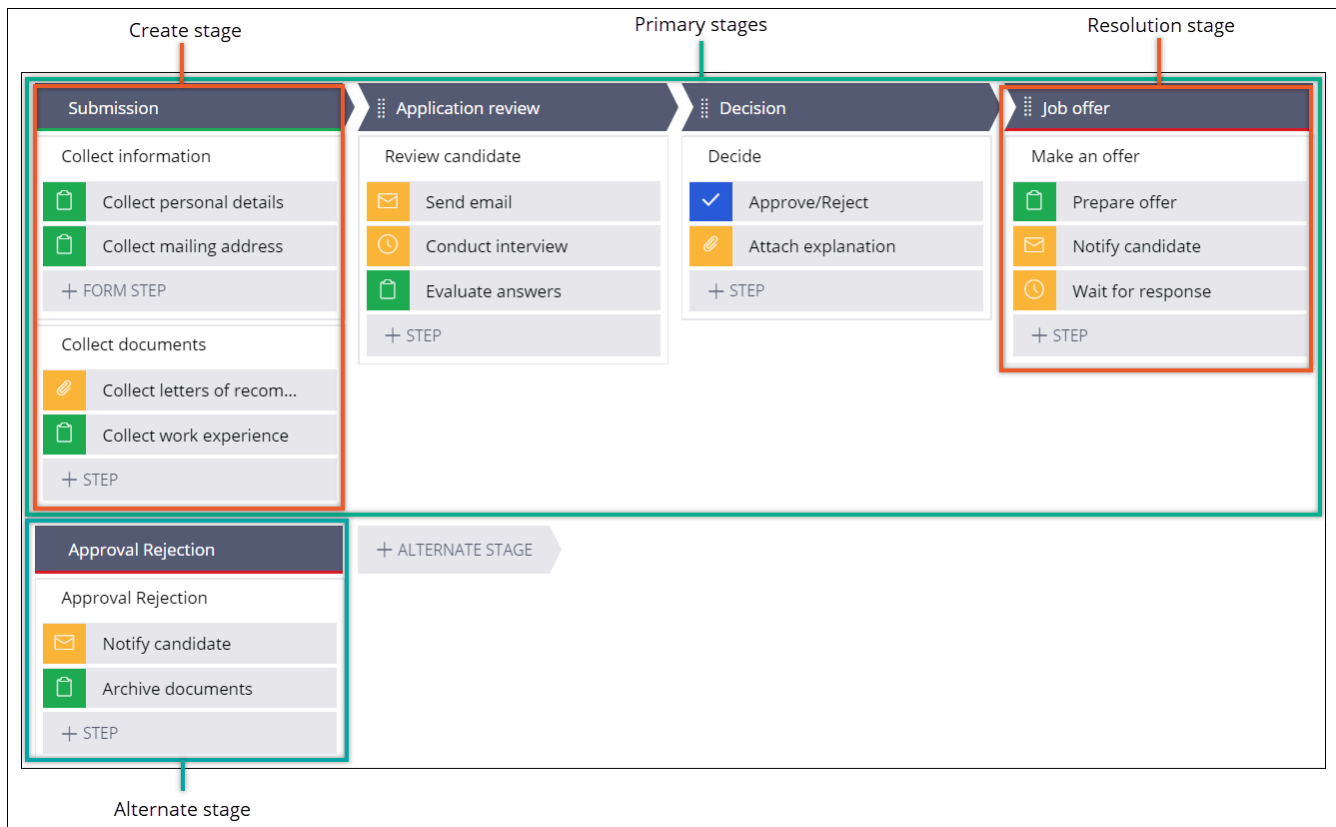
For example, if your business goal is to hire candidates for a certain role, you can divide the case into the following stages that represent the expected path of your case:

- Submission
- Application review
- Interview
- Decision
- Job offer

Stages define the top level organization of tasks that your business processes require to reach resolution. Stages represent milestones in your cases, such as finalizing a phase in a business process. For example, in a process for reviewing job applications, one stage visualizes actions in conducting a job interview and evaluating the documents that a job candidate provides. Another stage might consolidate tasks for making a decision about accepting or rejecting a candidate. Use nouns and noun phrases for stage names.

For improved flexibility and accelerated case resolution, multiple users can work on tasks within one stage. To ensure that you can resolve your business process in an alternative way when an exception occurs, for example, if the candidate does not meet all the requirements, you can create an alternate Approval Rejection stage.

The following figure presents a case life cycle with primary stages and an alternate stage:



Case Designer displaying primary stages and an alternate stage

Stages provide modularity in your case life cycle, which improves the flexibility of your application. You can configure a case to enter a stage only after meeting specified conditions, so that users interact only with actions that are relevant in a specific scenario. Additionally, a case can re-enter a stage, for example, so that a user can change their personal details. As a result, users avoid having to reprocess an entire case life cycle from the beginning.

To speed up case creation, when you create a new case type, the Case Designer provides you with a default Create stage. The Create stage includes views and actions in which users interact before they submit a case. Consequently, you can clearly visualize what data users require to be part of the case pre-processing. For example, in the Create stage, you can configure a view that collects personal information from a job candidate.

The resolution stage in your case type defines case behavior after reaching the end of the case life cycle. When you configure the resolution stage you define a final case status as well as whether any child cases also reach resolution at the same time. Case Designer marks the resolution stage with a red bar. A case type can have more than one resolution stage if you define an alternate path for your business process. For example, in a case type for reviewing applications from job candidates, if a case reaches resolution through the primary stages, the resolution status changes to `Resolved-Completed`. If the case ends on an alternate path, the resolution status changes to `Resolved-Rejected`. Use case statuses to clearly communicate case outcomes.

## Naming convention for stages

Ensure that names for stages clearly convey the meaning of a stage and are consistent throughout your case types. Typically, use nouns for stage names, for example, Submission, Review, and Decision. Avoid using verbs that describe an action or a status, for example, Order Placed, Shipped, Account Paid, or Closed.

Populate your case life cycle with stages by completing the following tasks:

The following video shows how to add a stage to your case type and what options you can configure for a stage:

[https://players.brightcove.net/1519050010001/default\\_default/index.html?videoId=6096759593001](https://players.brightcove.net/1519050010001/default_default/index.html?videoId=6096759593001)

---

Creating a Microjourney for customer success *(on page 329)*

Setting service-level agreements (SLAs) for stages, processes, and steps *(on page 330)*

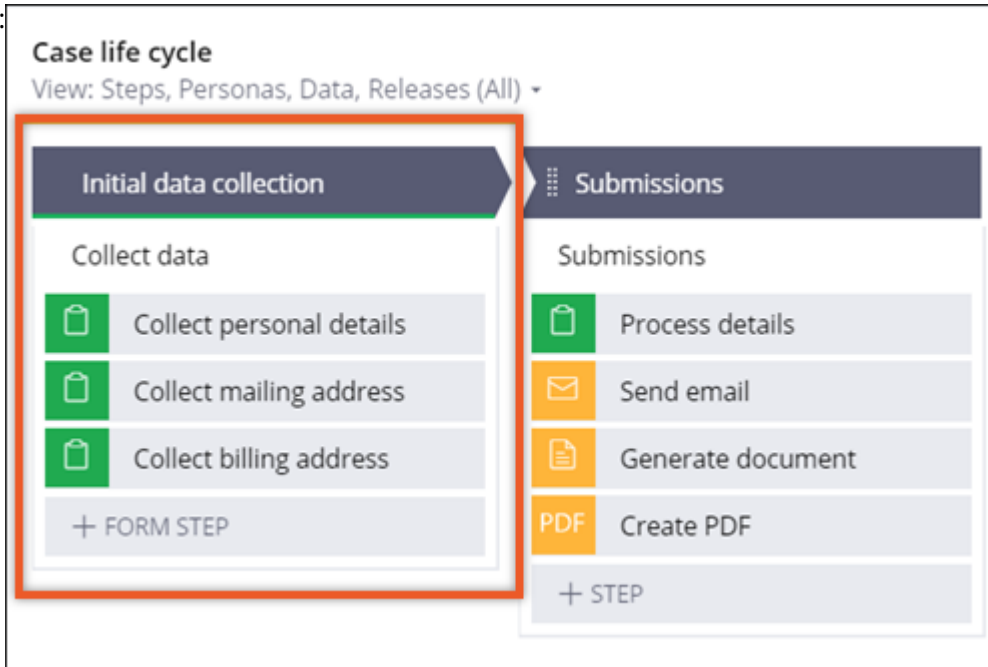
## The Create stage

To speed up case creation and clearly visualize the elements that users interact with before case processing starts, your case life cycle includes a default Create stage every time you create a new case type.

When you create a new case type, your case life cycle includes the Create stage by default. The Create stage holds a view and actions that users interact with to provide initial data before case processing starts. For example, in an application for reviewing loan requests, you can define a view that displays fields to collect the customer's personal details and their requested loan amount.



Case Designer marks the Create stage with a green bar, as in the following example:



Case Designer displaying a case type with a renamed Create stage

To learn more about the Create stage, explore the following articles:

- [The Create stage FAQ \(on page 49\)](#)
- [Adding the Create stage to existing case types \(on page 54\)](#)
- [Technical considerations for the Create stage in existing case types \(on page 55\)](#)

### Related information

[Processes in a case life cycle \(on page 66\)](#)

[Case types \(on page 28\)](#)

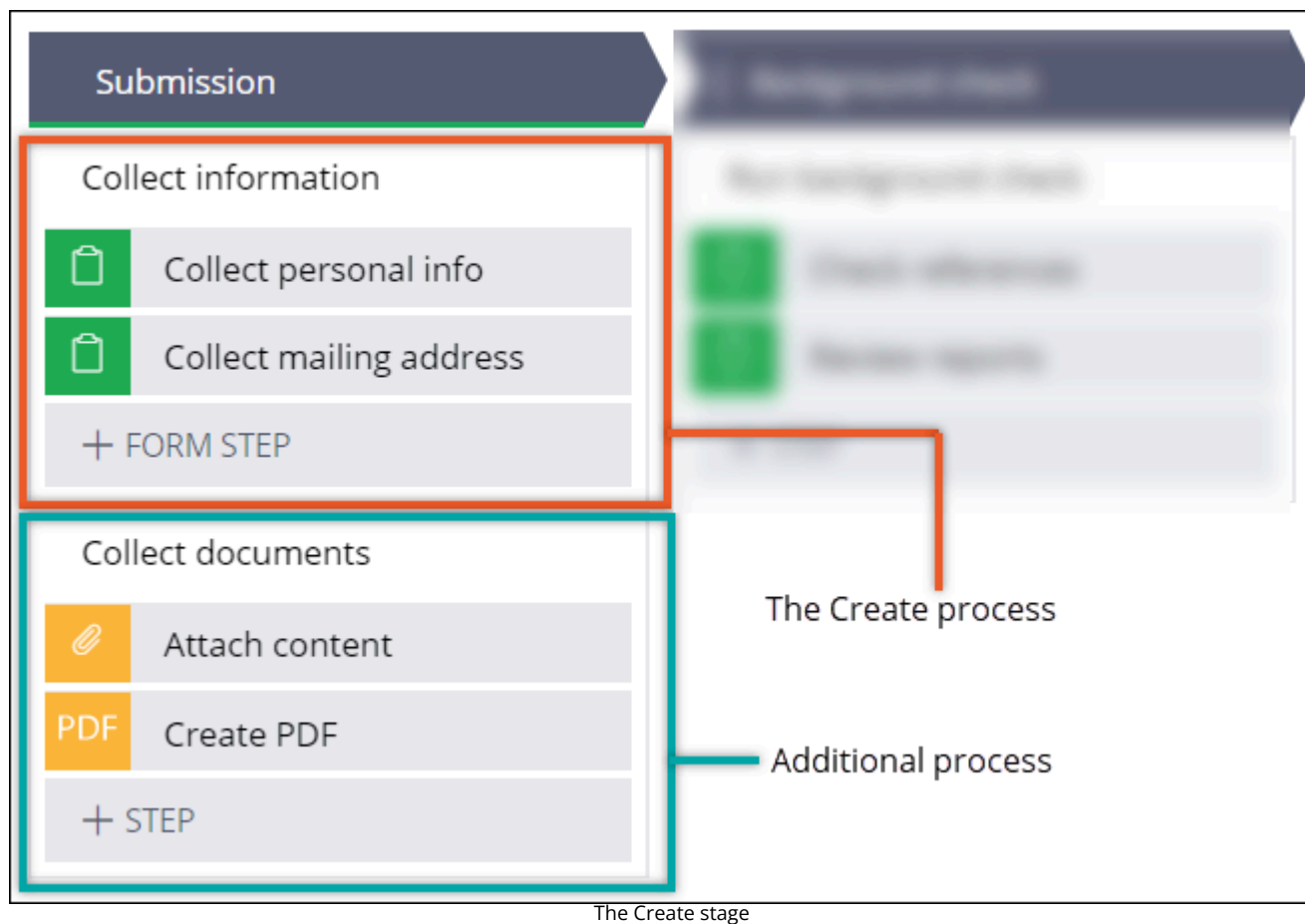
## The Create stage FAQ

The Create stage captures initial data before case processing starts, so that case creation experience across channels is more model-driven and intuitive. By understanding how the Create stage works, you can use it in a way that is the most efficient and appropriate for your business scenario.

## What does the Create stage include?

The Create stage includes a default Create process with a multistep form. You can modify the stage to meet your business needs by adding processes and steps. For greater flexibility, you can also rename the Create stage and the Create process to unique names that fit your business scenario.

The following figure shows a Create stage that is renamed to “*Submission*”, which includes the default Create process renamed to “*Collect information*”, with a multistep form. The Create stage in the example also includes an additional “*Collect documents*” process with two automated steps:



For more information about stages and processes, see [Case life cycle elements \(on page 43\)](#).

## When does the processing of cases with the Create stage start?

The system creates a new case in the database right after an end user of your application opens the screen that displays the Create stage. The Create stage is related only to case creation, and actual case processing starts when the case moves to the stage that is next after the Create stage.

## What happens if a user abandons a case in the Create stage?

If a user abandons a case in the Create stage, for example, when a user navigates away from the browser tab with the case and the browser session ends, the abandoned case remains with the `New` status. If the user cancels the case, the case automatically moves to the `Resolved-Canceled` status, and worklists

and work queues omit the case to better facilitate workload management. Maintaining such cases in your system helps you analyze and understand the reasons why users might discard cases at an early stage. If you do not want to maintain cases with the `Resolved-Cancelled` status, create an appropriate archive and cleanup policy.

## Can I configure a case type with the Create stage to be a temporary case at run time?

Cases that include the Create stage are persistent by default, which means that you cannot create a temporary case from a case type that includes the Create stage.

The following table shows differences between temporary cases and case types with the Create stage:

Case types with the Create stage	Temporary cases
Information about how users interact with a case from the moment of case creation	No information about user interaction with a case before the case becomes persistent
Support for all capabilities that are available for case types	No support for numerous capabilities, such as: <ul style="list-style-type: none"> <li>• audit trail</li> <li>• attachments</li> <li>• service-level agreements</li> <li>• child case types</li> <li>• routing assignments to specific users</li> </ul>
Persistent by default	Need to configure when the case becomes persistent

Consequently, only case types with the Create stage provide all information about user interaction and support all available features.

If your business scenario absolutely requires creating temporary cases, you can create a temporary case on a case type rule form in Dev Studio. However, temporary cases are a deprecated feature and creating such cases in Pega Platform version 8.5 and later might have update impact in the future. For more information, see [Creating temporary cases on a case type rule form \(on page 210\)](#).

## How does the Create stage work with creating a case by email?

When creating a case by email, at run time, your application omits the default Create process. The purpose of the Create process is to capture the initial data that is required before submitting a case, but because

an email starting a case meets the same requirement, the application skips the Create process. For more information, see [Enabling creation by email of top-level cases \(on page 226\)](#).

## Can I use the Create stage for different channels?

To reach wider audiences, you can create a channel-specific process inside the Create stage, for example, a process for users who interact with your application through a chat bot.

## Do all my case types include the Create stage by default?

All case types that you create in Pega Platform version 8.5 and later have the Create stage by default. Case types that you migrate from Pega Platform version 8.4 and earlier do not include the Create stage. You can add the Create stage to your migrated case types to take advantage of all the possibilities that the Create stage offers. For more information, see [Adding the Create stage to existing case types \(on page 54\)](#).

## Can I remove the Create stage from my case type?

The Create stage is a mandatory element for collecting information before a user submits a case, so you cannot remove the Create stage from a case life cycle.

## Can I configure entry validation for the Create stage?

A case always enters the Create stage, so you do not need to configure entry validation. If you want to configure validations upon case creation, configure validations as part of the view for the first assignment in the Create stage. For more information, see [Validating field values on a form \(on page 183\)](#).

If the Create stage in your business scenario includes no assignments, configure validation for the first stage after the Create stage. For more information, see [Defining the entrance criteria for a stage \(on page 58\)](#).

## Can I route an assignment in a Create stage to a work queue?

The main purpose of the Create stage is to collect initial data before case processing, so configure routing of the assignments in the Create stage to a current user. If your business scenario requires routing assignments to a work queue or a user different than a current user, configure routing logic on a stage that follows the Create stage. For more information, see [Managing work across your team \(on page \)](#).

## Can I configure parallel processes in the Create stage?

Yes, you can configure parallel processes in the Create stage. As a best practice, configure your application to route parallel processes to a current user, because the purpose of the Create stage is to collect data before case processing begins. For more information, see [Adding a parallel process to a stage \(on page 69\)](#).

## What harness does the Create stage include?

The Create stage includes the *pyCreate* harness for assignments. Other stages in a case type include the *Perform* harness for assignments.

## Can a user navigate back to the Create stage?

No, a case can only enter the Create stage once.

## Does the Create stage include the starting process?

Case types that include the Create stage are independent of the *pyStartCase* or any other starting process during case creation. As a result, your application is more granular and easier to maintain because you do not provide the starting process when you configure various functionalities, such as email instantiation of cases. All the configuration that in the previous versions of Pega Platform corresponded to the *pyStartCase* process, you now configure on the Create stage.

Only case types that you migrate from Pega Platform version 8.4 or earlier and that do not include the Create stage rely on the starting process.

## How can I pass parameters to a new case upon case creation if my case types are independent from the *pyStartCase* starting process?

You can still pass parameters when users create new case types by configuring the Create work action.

## If my child case type includes the Create stage, how do parent and child case types interact?

When a child case starts from the parent case, the child case enters the Create stage and waits for the user interaction with the first assignment in the Create process, similarly to other case types with the Create stage. If your business scenario requires an application to skip the Create process in a child case, define conditions to start the Create process only under specific circumstances. For more information, see [Conditionally starting a process \(on page 74\)](#).



## Can I start a case on the stage that follows the Create stage?

You can leave the Create stage unpopulated and start processing on the stage that follows the Create stage. However, your system still creates a new case when an end user interacts with the Create stage at run time. As a best practice, use the Create stage to collect initial data before case processing starts.

## How does case creation occur when users create cases through V1 and V2 DX APIs?

A case always enters the Create stage upon case creation through any channel, also through DX APIs. For greater flexibility, you can use the *pxCreateFromChannel* field to conditionally start a specific process that you want to run within the Create stage. The default value of the *pxCreateFromChannel* field is *web*. You can pass a different value as part of the *x-Origin-Channel* when you invoke the DX API. For more information, see [Pega Digital Experience \(DX\) API Overview \(on page 100\)](#).

---

[Enabling field-level auditing \(on page 212\)](#)

[Configuring case creation \(on page 223\)](#)

[Completing work on time \(on page 324\)](#)

[Attaching content to a case \(on page 101\)](#)

[Managing work across your team \(on page 100\)](#)

[Persisting temporary cases \(on page 134\)](#)

[Creating temporary cases \(on page 210\)](#)

## Adding the Create stage to existing case types

To fully experience all of the possibilities that the Create stage brings to your case life cycle, add the Create stage to your existing case types after you update your system to Pega Platform 8.5. With the Create stage, you clearly communicate the actions that users perform and which view elements users interact with before submitting a case for processing. For example, in an application for reviewing job requests, you can configure the Create stage to display a view with fields to gather a candidate's personal details and required documents.

When you update your system from Pega Platform 8.4 or earlier, you can include the Create stage in your existing case types. By adding the Create stage to your case life cycle, you clearly visualize the views and actions that users interact with before submitting a case. If you choose not to add the Create stage, your application is still fully backward compatible. After the update, your case types work correctly and rely on the *pyStartCase* starting process. Your case types become independent of the starting process only after you add the Create stage. Adding the Create stage is optional and requires additional configuration, however, the Create stage helps you define your case types and develop your application faster. To fully understand the impact of adding the Create stage to your existing case types, analyze the technical



considerations. For more information, see [Technical considerations for the Create stage in existing case types \(on page 55\)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the header of the **Case type** work area, click **Actions > Add create stage**.
4. In the **Add create stage** window, analyze the changes that occur after you add the Create stage, and then click **Submit**.
5. Click **Save**.

If you configured your case type to skip the Create view, the system marks the first stage in your case life cycle as the Create stage, and displays the view associated with the stage as the Create view. If you configured your case to display the Create view, the system adds a new Create stage as the first stage, and you need to define the Create view. For more information about the characteristics of case types with the Create stage, see [The Create stage \(on page 48\)](#).

#### Related information

[Processes in a case life cycle \(on page 66\)](#)

[Case types \(on page 28\)](#)

## Technical considerations for the Create stage in existing case types

To ensure that your application works correctly after you add the Create stage to your existing case types, consider the actions that you need to take in different technical scenarios.

## The Create stage in existing case types

To understand how the Create stage can impact your existing case types, consider the following scenarios:

- If you configured your existing case type to skip the Create view when users create a case, Case Designer marks the first stage in your case type as the Create stage by adding a green bar to the stage header.
- If you configured your existing case types to display the Create view, Case Designer adds the Create stage as the first stage in your case type.
- Your case types are independent of the *pyStartCase* or any other starting process. As a result, any security settings or other configurations that you define on the starting process rule are inapplicable to the case type.



- If you configured your case type to create temporary cases, the option is not applicable after adding the Create stage. All the case types that include the Create stage create persistent cases.
- When you configure the Create stage in a case type, the system immediately creates a case every time you enter the Create stage. If you do not want to include cases that are in the Create stage or have the `Resolved-Canceled` status in your dashboards or reports, update the relevant report definitions to filter out such cases.

To decide which scenario applies to your needs, consider the configuration of the starting process in your application:

## The case type rule has only *pyStartCase* configured as a starting process without modifications

Consider the following circumstances:

- You need to configure the view for the Create step in the newly added Create stage.
- The only data transform that works upon creation of a case with the Create stage is *pyDefault*. If you configured another data transform to work upon a case creation in your existing case type, Pega Platform automatically changes the data transform to *pyDefault*. To apply your custom configurations, call your data transform from the *pyDefault* data transform or configure a process in the Create stage to transform data.
- If you configured any validations on case creation, you need to reconfigure the validations to apply to the next stage after the Create stage.

## The case type rule has only *pyStartCase* configured as a starting process with modifications or the case type rule has many starting processes configured

Consider the following circumstances:

- You need to configure the view for the Create step in the newly added Create stage.
- You need to move any pre-processing or post-processing logic from the *pyStartCase* starting process to the Create stage.
- The only data transform that works upon creation of a case with the Create stage is *pyDefault*. If you configured another data transform to work upon case creation in your existing case type, Pega Platform automatically changes the data transform to *pyDefault*. To apply your custom configurations, call your data transform from the *pyDefault* data transform or configure a process in the Create stage to transform data. For more information, see [Data transforms \(on page 10\)](#).
- If you configured any validations on case creation, you need to reconfigure the validations to apply to the next stage after the Create stage.





## Related information

[Processes in a case life cycle \(on page 66\)](#)

[Case types \(on page 28\)](#)

## Creating a primary stage

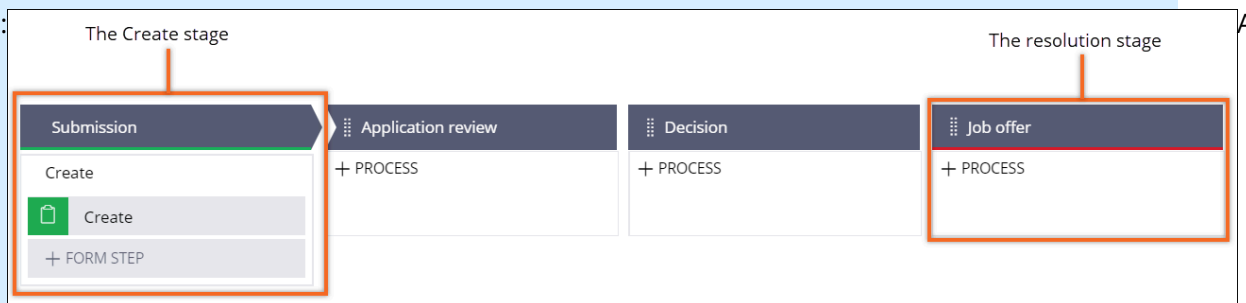
Represent main phases of your business process by creating primary stages in a case life cycle. By creating stages, you can ensure that your work is divided into logical phases, so that you can track and resolve the tasks more conveniently. Primary stages visualize the main phases of your business process. For example, for a Review job application case type, you can create stages, such as Submission, Application review, Decision, and Job offer.

When you create a new case type, the system automatically creates the first stage, which is the Create stage. For more information, see [The Create stage \(on page 48\)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, click **Stage**, and then enter a unique name in the text box.  
By default, the system creates a stage for each case life cycle that you define.
5. In the **Stage** property pane, on the **General** tab, define the transition after this stage is complete:
  - To move to the next stage, select **Automatically move to next stage**.
  - To let the customer service representative decide which stage the case enters, select **Wait for a user action**.
  - To close the case, select **Resolve the case**, and then define the resolution status and options.
6. Click **Save**.

The following figure shows a case type with four empty stages. A green bar marks the Create stage, and a red bar marks the resolution

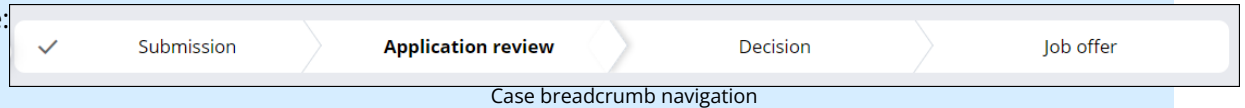
stage:



At run time, a breadcrumb navigation of a case displays names

that you associate with the stages, as shown in the following

figure:



[Case life cycle elements \(on page 43\)](#)

[Creating an alternate stage \(on page 58\)](#)

[Processes in a case life cycle \(on page 66\)](#)

## Creating an alternate stage

Ensure that you can resolve a business process when an exception occurs by adding an alternate stage to the case life cycle. For example, you can use an alternate stage to process cases that managers reject.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. Create an alternate stage:
  - To create a new alternate stage, in the **Case life cycle** section, click **Alternate stage**.
  - To convert a primary stage to an alternate stage, on the menu next to the stage name, click **Set as alternate stage**.
5. Decide how the case proceeds after all processes in this stage are complete:
  - To let the case worker decide which stage the case enters after this stage, in the stage property pane, select **Wait for a user action**.
  - To resolve the case, in the stage property pane, select **Resolve the case**, and then configure details of case resolution. For more information, see [Configuring a case resolution \(on page 62\)](#).
6. Click **Save**.

[Case life cycle elements \(on page 43\)](#)

[Creating a primary stage \(on page 57\)](#)

[Processes in a case life cycle \(on page 66\)](#)

## Defining the entrance criteria for a stage

Validate case properties, attachments, and conditions to determine whether a case can enter a specific stage as it moves through the case life cycle. For example, you can check whether a vehicle loan case has an attached credit score before sending the case to a finance manager for approval.



Use the following techniques to define entrance criteria for a stage:

- [Stages in a case life cycle \(on page 46\)](#)
- [Case life cycle elements \(on page 43\)](#)
- [Creating a primary stage \(on page 57\)](#)
- [Attachment types \(on page 458\)](#)
- [Controlling access to case attachments \(on page 457\)](#)
- [Categorizing case attachments \(on page 457\)](#)
- [About Validate rules \(on page \)](#)

## Setting required attachments for stage entry

Maximize efficiency and speed up the case resolution process by verifying that the user provides required, relevant documentation and correspondence before a case enters the next stage.

For example, in a case for a car insurance claim, you can verify that the policyholder attaches photographs of their damaged car before the case enters the stage in which the insurance company validates the policyholder's claim.



**Note:** You cannot define entrance criteria for the Create stage. For more information, see [The Create stage \(on page 48\)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Workflow** tab, click a stage name.
3. In the **Stage** property pane, click **Validation**.
4. In the **Attachments required for stage entry** section, click **Add required attachment** to display the list of attachment categories that your case type supports.
5. Select an attachment category from the list.
6. Click **Save**.

At run time, users need to provide at least one attachment in each of the required attachment categories, otherwise the application displays an error.

## Validating field values for stage entry

Ensure that a case contains relevant data before it enters a specific stage by validating the field values that users provide. By validating field values for a stage entry, you prevent the case from proceeding to the next stage without the relevant information. If the value that the user enters into a form at run time meets



the condition in a validate rule, the application displays an error message. For example, you can ensure that a job applicant provides their work experience before the job application case enters a Decision stage in which a manager approves or rejects the application. If the work experience fields are empty when the case reaches the Decision stage, the application displays an error message and the case stays in the current stage until the job applicant completes the necessary fields.



**Note:** You cannot define entrance criteria for the Create stage. For more information, see [The Create stage \(on page 48\)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Data model** tab, click **Validations**.
4. Select the cell that corresponds to the row of the field and the column of the stage for which you want to perform validation:
  - To define validation for a single-value field, click **Add entry validation**.
  - To define validation for a data relationship field, click **Click to configure**, and then click **Add entry validation**.
5. In the **Stage entry validation** dialog box, define the validation conditions:

#### Choices

##### Define a validation condition for a field

#### Actions

- a. In the list of values, select **Fields**, and then select the name of the field that you want to validate.  
By default, the system provides the field that you select in step 4 [\(on page 60\)](#).
- b. In the comparator list, select the test that you want to perform on the field.
- c. In the value field, enter or select a value to compare against the user input.

Choices	Actions
<b>Apply an existing when condition as a validation condition</b>	<ul style="list-style-type: none"> <li>a. In the list of values, select <b>When conditions</b>, and then select the name of the when condition that you want to apply.</li> <li>b. In the comparator list, select the test that you want to perform on the when condition.</li> </ul>

6. **Optional:** To create a condition with multiple validation parameters, add a logical operator to the condition:

Choices	Actions
<b>The condition passes when all properties meet the criteria</b>	<ul style="list-style-type: none"> <li>a. Click <b>Add a row</b>.</li> <li>b. In the operator list, select <b>and</b>.</li> <li>c. In the list of fields, select the name of the field that you want to validate or the name of the when condition that you want to apply.</li> <li>d. In the comparator list, select the test that you want to perform on the field or on the when condition.</li> <li>e. In the value field, enter or select a value to compare against the user input. The <i>Select values</i> option lists the fields and values that you can use in the condition.</li> </ul>
<b>The condition passes when any of the properties meet the criteria</b>	<ul style="list-style-type: none"> <li>a. Click <b>Add a row</b>.</li> <li>b. In the operator list, select <b>or</b>.</li> <li>c. In the list of fields, select the name of the field that you want to validate.</li> <li>d. In the comparator list, select the test that you want to perform on the field or on the when condition.</li> </ul>

Choices	Actions
	<p>e. In the value field, enter or select a value to compare against the user input. The <code>Select values</code> option lists the fields and values that you can use in the condition.</p>

7. In the **Stage entry validation** window, in the **Then display error message as** field, enter the text to display when user input meets the validation conditions at run time.
8. **Optional:** To select additional stages for which you want to perform the validation, in the `To validate stage entry for the stages` parameter, select the stages. By default, the system selects the check box for the stage that you select in step 4 (on page 60).
9. Click **Submit**.
10. **Optional:** To define more validation conditions, in the cell in which you want to add more conditions, click **Add another validation**, and then repeat steps 5 (on page 60) through 9 (on page 62).
11. Click **Save**.

If you configure validation for a single-value field, the cell displays a message for the condition. If you configure a validation for a data relationship field, the validation matrix cell displays `Validated: Click to view` for the top-level field and a message for the condition for the embedded fields.

## Configuring a case resolution

Configure a resolution method in a stage to control the final status and processing that occurs when a case reaches the end of its life cycle. For example, you can configure a Loan request case type to have two possible resolution statuses: `Resolved-MoneyDisbursed` and `Resolved-InvalidClaim`. To reach the `Resolved-MoneyDisbursed` status, you configure the case to follow the primary stage sequence. To reach the `Resolved-InvalidClaim` status, you redirect the case to an alternate stage.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. Click the stage at which you want to resolve the case. Typically, the last primary stage in the life cycle or an alternate stage resolves a case.
4. In the **Stage** property pane, on the **General** tab, click **Resolve the case**.
5. **Optional:** To change the default final status for the resolved case, select the new status:

- To use an existing status, in the **Resolution status** field, press the Down arrow key, and then select a status value.
  - To create a new status, in the **Resolution status** field, enter a unique name by using the format `Resolved-[your status name]`.
6. **Optional:** To clean up dependencies when the current case reaches its resolution, specify the cleanup scope:
- a. **Optional:** To delete any open assignments from the previous stages, select the **Delete open assignments** check box.
  - b. Select the **Resolve open child cases** check box.  
If you clear the check box, the child cases continue their execution.
7. **Optional:** To change the default final status for the resolved child cases, select the new status:
- To use an existing status, in the **Resolution status of child cases** field, press the Down arrow key, and then select a status value.
  - To create a new status, in the **Resolution status of child cases** field, enter a unique name by using the format `Resolved-[your status name]`.
8. Click **Save**.

When all steps in the stage reach completion, the case and its open child cases change status to the new values. By default, when a case reaches its resolution, the current case is set to `Resolved-Completed`, and open child cases are set to `Resolved-Unspecified`.

[Stages in a case life cycle \(on page 46\)](#)

[Case life cycle elements \(on page 43\)](#)

[Identifying case types elements \(on page 9\)](#)

[Building case types \(on page 20\)](#)

## Defining conditions for skipping a stage

Ensure that cases move to the next stage in the sequence when the current stage is not relevant, by defining the conditions that cause a case to skip a stage. When a customer service representative (CSR) can skip irrelevant stages, a case reaches its resolution faster.

For relevant training materials, see the [Skipping a process or a stage](#) module on Pega Academy.


At run time, your application evaluates the When condition or expression before the case enters the stage. If the value is true, the case skips the processes in the stage and moves to the next stage in the sequence.

At run time, your application evaluates the condition before the case enters the stage. If the value is true, the case skips the processes in the stage and moves to the next stage in the sequence.



1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, click a stage name.
5. In the case working area, open the **Validation** tab.
6. In the **Stage** property pane, on the **General** tab, define the condition for skipping the stage:

Choice	Steps
<b>Never skip a stage</b>	a. In the <b>Skip stage</b> list, select <b>Never</b> .
<b>Use a custom condition</b>	a. In the <b>Skip stage</b> list, select <b>Custom condition</b> . b. Click the <b>Configure conditions</b> icon. c. In the <b>Configure condition</b> window, define a condition that starts the process. d. <b>Optional:</b> To add more conditions, click <b>Add a row</b> , and then repeat substep <a href="#">6.c (on page 64)</a> . e. <b>Optional:</b> To group the conditions, select operators in the list. f. Click <b>Submit</b> .
<b>Use an existing condition</b>	a. In the <b>Skip stage</b> list, select <b>Existing condition</b> . b. In the list of conditions, select a condition, and then define whether the process skips when the condition is true or false.

 **Note:** You cannot skip the Create stage and last stage in the sequence.

7. In the **Stage** property panel, on the **General** tab, define a condition for skipping the stage:



Choice	Steps
<b>Never skip a stage</b>	a. In the <b>Skip stage</b> list, select <b>Never</b> .
<b>Use a When rule</b>	a. In the <b>Skip stage</b> list, select <b>When rule</b> . b. In the text field, press the Down arrow key, and then select a when rule that stores the conditions for skipping the stage.
<b>Use an existing expression</b>	a. In the <b>Skip stage</b> list, select <b>Expression</b> . b. In the text field, press the Down arrow key, and then select an expression that stores the conditions for skipping the stage.

**Note:** You cannot skip the Create stage and the last stage in the sequence.

8. Click **Save**.

At run time, the system evaluates the condition and skips the stage if the value meets the condition.

[Case life cycle elements \(on page 43\)](#)

[Stages in a case life cycle \(on page 46\)](#)

[The Create stage \(on page 48\)](#)

## Removing a stage


Deliver a flexible application that you can conveniently adjust to dynamically changing circumstances by removing stages that are no longer relevant to your business requirements. As a result, you save time because you can edit an existing case type instead of creating a new one from scratch. For example, if the process for hiring new employees in your organization changes and one of the stages in your hiring case type, such as running a background check, is no longer required, you can simply remove the stage to reflect the change and adjust the existing case type to the new hiring process.

You can delete primary stages that visualize the happy path of your business process, as well as alternate stages that correspond to actions that occur if your business process diverts from the primary path.

You cannot delete the Create stage that every case type includes by default. Removing a stage does not



delete any resources that you connect to the stage, such as service-level agreements or notifications. The resources are present in your system but do not participate in case processing anymore. You can reuse these resources in other elements of your application.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. Hover over the stage name.
3. In the stage header, click the **More**  icon, and then click **Delete stage**.
4. Click **Save**.

---

[The Create stage \(on page 48\)](#)

[Empowering knowledge workers \(on page 348\)](#)

[Case life cycle elements \(on page 43\)](#)

[Finding rules by type \(on page \)](#)

## Processes in a case life cycle

Processes in stages organize related tasks in your business case. With processes, you can control the order of events in your case, in addition to who performs the work and in what manner. For example, for the Application review stage of a recruitment case, you can add the Conduct interview and Review collected information processes.

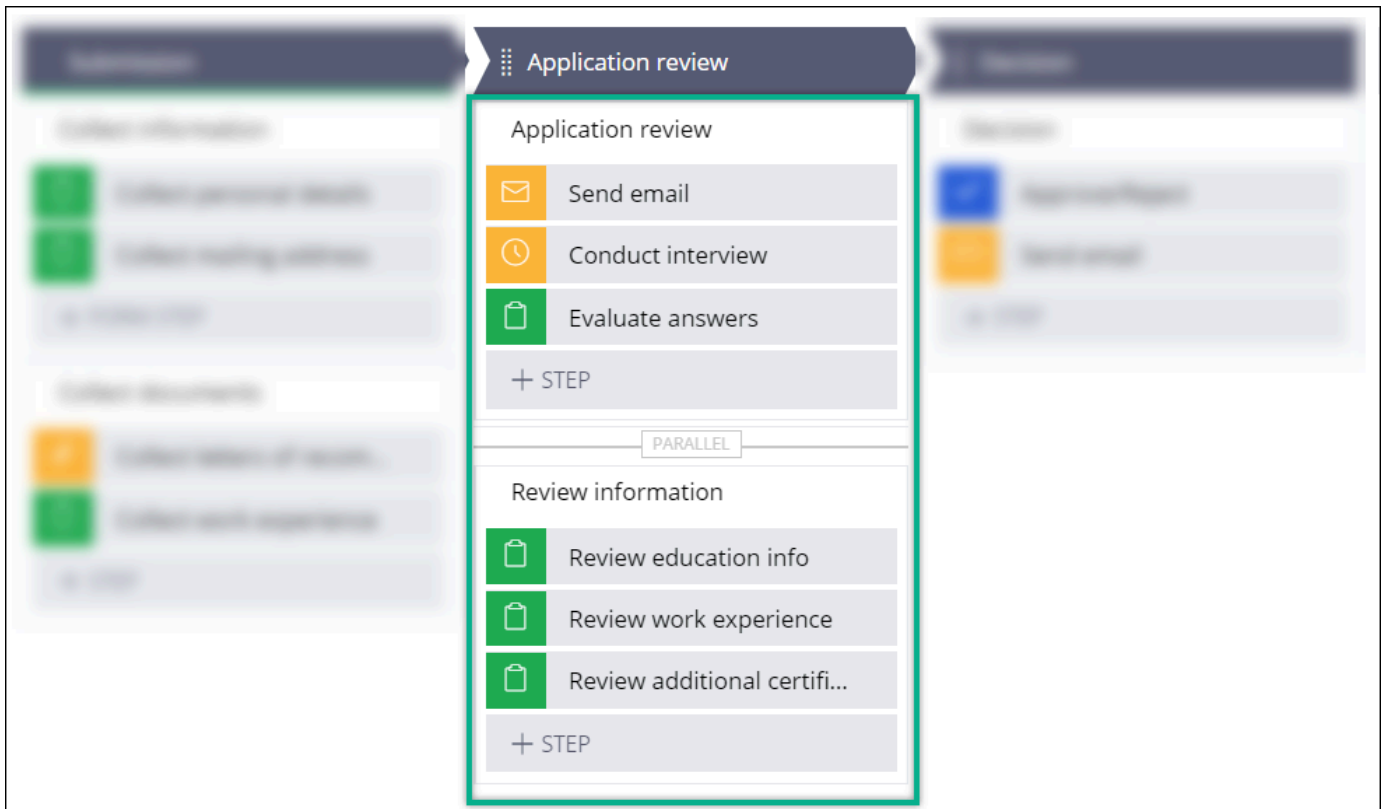
When you add processes to a case type, you organize related tasks in a logical way, instead of having a list of loosely connected assignments. You also define an order of case events, so that a case can only move to the next process after completing the steps in the current process. For example, when reviewing a job candidate, a process that includes a job interview can only start after a process for collecting all required documents is complete.

The following image shows two sequential processes in a stage: Collect information and Collect documents. A case can enter the second stage only after the first stage is complete.

## Sequential processes in a stage

To resolve cases faster, you can also create multiple parallel processes within one stage. As a result, users can work simultaneously during the case cycle, for example, while one HR worker verifies documents collected from a job candidate, another HR worker runs a background check. For an improved working experience, you can define conditions for when a process starts, so that users only interact with actions that are relevant to specific scenarios, such as starting a process of additional interviews only when this action is relevant. By creating processes, you also speed up application development, since you can reuse processes in many different case types in your application. When you name processes, use verb and noun combinations.

The following figure shows an Application review stage that consists of the Conduct interview and Review collected information parallel processes:



Populate your case life cycle with processes by completing the following tasks:

- [Creating a Microjourney for customer success \(on page 75\)](#)
- [Removing a process from a stage \(on page 76\)](#)

## Adding a sequential process to a stage

Control the order of events in a case by adding a sequential process to a stage. A sequential process orders related actions that lead to the resolution of a case.

For example, if your business case is to review a job application, you can create a sequential process that includes the following tasks:

- Prepare interview questions
- Conduct the interview
- Prepare the candidate's assessment

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.

4. In the **Case life cycle** section, hover over a stage, and then select a process to add:
  - To add a new process, click **More > Add process > New process**.
  - To reuse an existing process, click **More > Add process > Process name**.
5. In a new text field, replace the default process label with a descriptive name.
6. **Optional:** To change the run-time order of a process, drag the process to a different position in the stage.
7. Click **Save**.

At run time, the process starts when all steps in the previous process are complete.

[Conditionally starting a process \(on page 74\)](#)

[Creating a stand-alone process \(on page 249\)](#)

## Adding a parallel process to a stage

Support your business process events that do not require a run-time order by adding a parallel process to a stage. Parallel processing speeds up case resolution as different case workers can complete parallel processes simultaneously.

For example, if your business goal is to review a job candidate, you can create two parallel processes: one for conducting a job interview, and one for reviewing documents that the candidate provides. A hiring manager can interview the candidate, while an HR worker reviews the documents.

For relevant training materials, see the [Parallel processing](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, hover over a stage, and then add a process:
  - To add a new process, click **More > Add parallel process > New process**.
  - To add a new multistep form, click **More > Add parallel process > New multistep form**.
  - To reuse an existing process, click **More > Add parallel process**, and then select the process in the list.
5. In the text field, replace the default process label with a descriptive name.
6. Click **Save**.
7. **Optional:** To prevent the process from running each time that a case reenters the stage, configure the settings:
  - a. Next to the name of the case type, click **Actions > Open**.
  - b. On the **Stages** tab, click the name of the stage that contains your process.



- c. In the **Automatically launched processes** section, clear the **Run on re-entry** check box next to the name of your process.
- d. Click **Save**.

At run time, the process starts when a case enters the stage.

[Conditionally starting a process \(on page 74\)](#)

[Creating a stand-alone process \(on page 249\)](#)

## Adding a multistep form to a stage

Break a single assignment that captures many fields into multiple focused and concise screens by adding a multistep form to a stage. You can add both a sequential process and a parallel process as a multistep form.

For example, you can ask users to enter their personal details on one screen, and then describe their medical condition on the next screen.

For relevant training materials, see the [Multi-step forms](#) topic on Pega Academy.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Workflow** tab, click **Life cycle**.
3. In the **Case life cycle** section, in a stage, click **More > Add process > New multi-step form**.
4. In the text field, replace the default process label with a descriptive name.
5. On the **General** tab, in the **Start process** section, define the condition to start the process:
  - To start the process when the case reaches this stage, select **Always**.
  - To determine the condition with a when rule, select **When rule**, and then select the rule.
  - To determine the condition with an expression, select **Expression**, and then select the expression.
  - To choose a custom condition that determines whether the process starts, select **Custom condition**, and then select the condition.
  - To reuse an existing condition, select **Existing condition**, and then select the condition.
6. In the properties panel, in the **Route to** list, identify the user or team that performs the task:
  - To assign the task to the user who last updated the case, select **Current user**.
  - To assign the task to another user in your application, select **Specific user**, and then configure the user parameters.

You can assign the task to a user by user name, a user by user reference, the reporting manager of the user who last updated the case, or a case participant.

- To assign the task to a team that shares a work queue, select **Work queue**, and then select a team.
- To assign the task to a worklist or a team that shares a work queue, select **Custom**, and then configure the team or worklist parameters.

For more information about configuring custom routing, see [Configuring custom routing logic for assignments \(on page 89\)](#).

- To automatically assign the task to a user with the appropriate availability, skill set, or workload, select **Use business logic**, and then click the **Configure business logic** icon.

For more information about configuring business-based routing, see [Assigning users automatically at run time with business logic \(on page 85\)](#).

7. In the **Template** section, select the navigation type in the multistep form by clicking **Change**, and then choosing the navigation template.

The default setting is horizontal navigation.

8. Add a step to the multistep form by clicking **Form step**.

For more information about configuring steps, see [Steps in a case life cycle \(on page 78\)](#).

9. In the text field, replace the default step label with a descriptive name.

10. Click **Save**.

The following figure shows a run-time presentation of a multistep form. A user that completes the form starts with completing the **Collect personal details** screen, and then moves to the **Collect medical information** screen.

Multistep form at run time

[Conditionally starting a process \(on page 74\)](#)

[Harnesses \(on page \)](#)

[Configuring complex processes \(on page 248\)](#)

[Creating a screen flow \(on page 276\)](#)

## Configuring advanced options for a multistep form

To ensure that you deliver an application that precisely meets your business requirements, define how a process that is a multistep form behaves by configuring advanced starting options. For example, you can set a status that a case enters when a process starts, and then define a user to complete the process.

When you edit a process on a flow canvas in Dev Studio, you have access to advanced configurations. On a flow canvas, you configure the initial behavior of the process by editing the Start shape. By default, every new process includes the Start shape.



**Note:** In Dev Studio, a multistep form is called a screen flow. For more information, see [Differences in naming between App Studio and Dev Studio \(on page \)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, hover over a multistep form in a stage, and then click the **Configure process** icon.
3. Click the **Open process** link, and then, on the canvas, double-click the **Start** shape.
4. Perform any number of the steps from [3 \(on page 277\)](#) to [11 \(on page 280\)](#) in [Creating a screen flow \(on page 276\)](#).

## Defining conditions for starting a process

Ensure that a process starts only when needed in a case by defining conditions for running the process. By starting processes conditionally, you save time because case workers, such as customer service representatives (CSRs), perform only relevant tasks.

For example, you can start a process of reviewing a job application only if the location of a candidate matches your current business needs.

For relevant training materials, see the [Skipping a process or a stage](#) module on Pega Academy.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, click a process.





3. In the **Process** property panel, on the **General** tab, define the conditions to start the process:

Choices	Actions
<b>Always start a process</b>	a. In the <b>Start process (otherwise skip)</b> list, select <b>Always</b> .
<b>Custom condition</b>	a. In the <b>Start process (otherwise skip)</b> list, select <b>Custom condition</b> .  b. Next to the <b>Start process (otherwise skip)</b> list, click the <b>Configure conditions</b> icon.  c. In the <b>Configure condition</b> window, select a condition, a comparator, and a value to compare with the condition.  d. <b>Optional:</b> To add more conditions, click <b>Add a row</b> , and then repeat the sub-step <a href="#">3.c (on page 73)</a> .  e. <b>Optional:</b> To group the conditions, select comparators from the list.  f. Click <b>Submit</b> .
<b>Existing condition</b>	a. In the <b>Start process (otherwise skip)</b> list, select <b>Existing condition</b> .  b. In the list of conditions, select a condition, and then define whether the process starts when the condition is true or false.

4. Click **Save**.

At run time, the system evaluates the conditions and starts the process when the conditions return true values.

### Related information

[Changing the path of a process \(on page 280\)](#)



## Conditionally starting a process


Create applications that flexibly adjust to dynamically changing scenarios by configuring conditions to start specific sets of actions. When processes in your case type start under specified conditions, you save time and resources because you can create one case type that is relevant in various situations, instead of providing and maintaining multiple case types. Additionally, you can focus your application development on a goal that you want to achieve rather than on fixed paths that might be difficult to predict.

Consider a case type with the goal of hiring a new candidate. The case type includes an `Interview` stage during which a manager interviews a job candidate. Additionally, if the candidate applies for a manager role, the company requires a background check. You can create a `Background check` process, and then define conditions to run the process only for candidates for manager roles. As a result, you create only one case type instead of creating two very similar case types.

You can define conditions to start a process both in App Studio and Dev Studio. In App Studio, you can reuse existing conditions or create new ones by using a condition builder, while in Dev Studio you can use a `When rule` or an expression to define when the process starts.

1. Open the process that you want to configure:
  - To configure the process in App Studio, in the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - To configure the process in Dev Studio, in the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, click the process that you want to configure.
3. In the **Process** pane, on the **General** tab, define conditions to start the process:

Choices	Actions
<p><b>Always start a process</b></p>	<p>In the <b>Start process (otherwise skip)</b> list, select <b>Always</b>.</p>
<p><b>Define a new condition in App Studio</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>Start process (otherwise skip)</b> list, select <b>Custom condition</b>, and then click the <b>Configure conditions</b> icon.</li> <li>b. In the <b>Configure condition</b> window, define a condition by selecting a property, comparator, and value to evaluate against the property at run time. As a property, you can select a field or a when condition. For more information about building conditions, see <a href="#">Defining</a></li> </ol>

Choices	Actions
	<p>conditions in the condition builder (<i>on page</i> ).</p> <p>c. <b>Optional:</b> To define more conditions, click <b>Add a row</b>, and then repeat step <a href="#">3.b (on page 74)</a>.</p> <p>d. If you have multiple conditions, specify whether all or any conditions need to evaluate to true at run time by selecting either <b>and</b> or <b>or</b> between the conditions.</p> <p>e. Click <b>Submit</b>.</p>
<p><b>Use an existing condition in App Studio</b></p>	<p>a. In the <b>Start process (otherwise skip)</b> list, select <b>Existing condition</b>.</p> <p>b. In the list of conditions, select a condition that you want to evaluate, and then decide whether the condition needs to evaluate to true or false to start the process.</p>
<p><b>Use a When rule in Dev Studio</b></p>	<p>a. In the <b>Start Process</b> list, select <b>When Rule</b>.</p> <p>b. In the text field, press the Down arrow key, and then select the When rule that you want to use.</p> <div data-bbox="938 1493 1463 1774" style="border: 1px solid #0070C0; background-color: #D9E1F2; padding: 10px; margin-top: 10px;"> <p><b>Tip:</b> You can also create a new When rule by clicking the <b>Open</b>  icon. For more information, see <a href="#">Creating a When rule (on page</a> ).</p> </div>

Choices	Actions
<p><b>Use an expression in Dev Studio</b></p>	<p>a. In the <b>Start Process</b> list, select <b>Expression</b>.</p> <p>b. In the text field, press the Down arrow key, and then select the expression that you want to use.</p> <div data-bbox="938 541 1463 720" style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p><b>Tip:</b> You can also create a new expression by clicking the <b>Build an expression</b> icon.</p> </div>

4. Click **Save**.

At run time, when a case reaches the process, an application evaluates the conditions and, based on the results, either starts or skips the process.

[Processes in a case life cycle \(on page 66\)](#)


[Displaying supporting processes conditionally \(on page 358\)](#)

[Displaying optional actions conditionally \(on page 355\)](#)

## Removing a process from a stage

To save time and resources, adjust your existing case types to changing business scenarios by removing processes that are no longer required. Modifying case types helps you efficiently use resources in your application and accurately respond to dynamic situations. In a sample scenario, if your organization no longer requires job candidates to deliver printed proofs of education, you can remove a process that creates the steps necessary to collect such documents.

By removing a process, you disassociate the process from a case. However, the process is still available in your application. You can reuse the process in the future, or in other case types.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. Hover over a process name in a stage.
3. Click the **Delete**  icon.
4. Click **Save**.

---

[Calling one process from another process \(on page 292\)](#)

[Creating a stand-alone process \(on page 249\)](#)

## Draft mode of case processes

When you add a process to your case life cycle, by default the process is in draft mode. Draft mode provides an option to test the case type before you advance with your application development, even if processes in your case life cycle contain configuration issues. For example, you can test how a Loan request case type behaves at run time even before you configure the views with fields that users use to provide information.

Every process that you add to your case life cycle is automatically put into draft mode. As a result, you can preview and run a case to check the run-time behavior, even if some of the processes contain errors. When you want to deploy your application in a production environment, to correctly migrate processes you need to turn draft mode off. Because you can add multiple processes to every case type in your application, App Studio and Dev Studio automatically turn draft mode off for every process that has no errors. Consequently, you save time as you avoid having to manually turn off draft mode in Dev Studio for every single process.

To ensure that the status of draft mode changes when you save your case type, if any of your processes contain errors, Case Designer displays a list of issues that you need to fix before deployment. The list navigates you to the relevant processes and provides a short description of each issue, as in the following examples:

Number of errors

Case type: Hire a candidate

Workflow Data model Views Settings

Case life cycle  
View: Steps, Personas, Data, Releases (All) -

Life cycle Optional actions

Submission Decision Approval + STAGE

Collect information

- Collect personal details
- Collect mailing address
- + FORM STEP

Collect documents

- Collect letters of recom...
- Collect work experience
- + STEP

Decision

- Approve/Reject
- + STEP

Inform candidate

- Send email
- Start onboarding
- + STEP

Warning about errors in processes

**Draft process warnings**

The following configuration issues need to be addressed before the processes can be toggled out of draft mode and published to production.

Issue	Stage	Process
Connector— View must be configured for Create (unable to open Flow Action)	Create	Create details

Window with a list of errors

## Related information

[Configuring case type settings \(on page 203\)](#)

[Configuring complex processes \(on page 248\)](#)

[Managing case types \(on page 315\)](#)

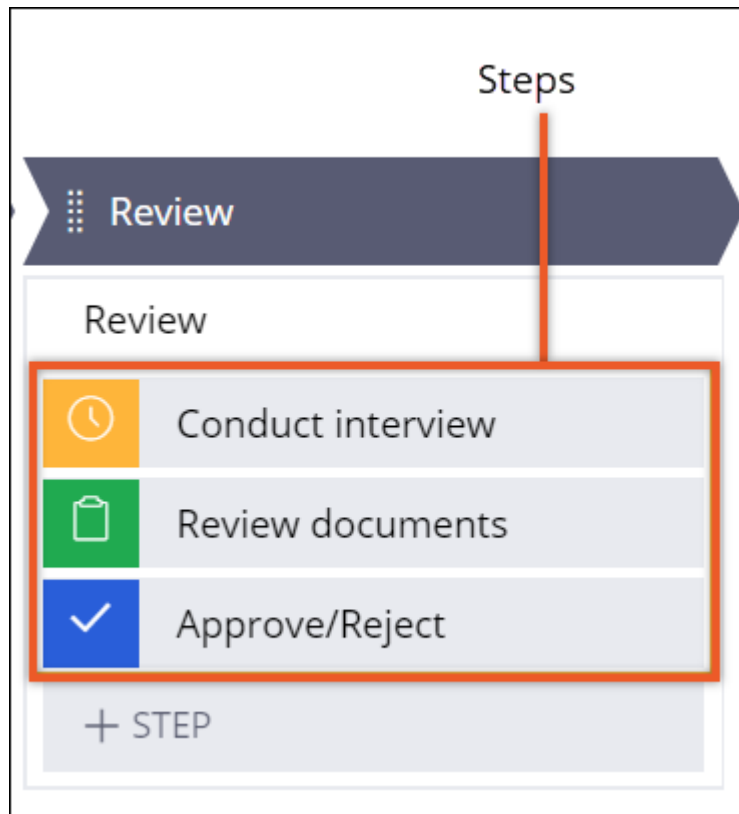
## Steps in a case life cycle

A step in your case life cycle represents a single assignment in your business process. By adding a step, you define a task that a user, your application, or an external application performs to move a case closer to resolution.

A step is the most granular element of your business process and represents one unit of work in your case. A step can be a user action, such as collecting information from a customer, or an automated action that an application performs, such as sending an email after a business process reaches a resolution. To save resources and speed up application development, you can select from a wide variety of automated steps. For greater flexibility, many steps offer additional configuration options, so that you can meet your precise business needs. For example, in a step to send emails, you specify a receiver or a group of receivers, a subject, and a message content. Because steps represent single actions, use verb and noun combinations to name steps.



The following figure presents a set of related steps:



Visualization of actions in a business process

When you visualize your work by adding steps to a case life cycle, you clearly communicate what actions your business process requires to reach a resolution, and you organize your work in the most efficient way.

Creating a Microjourney for customer success (on page 66)  
[Processes in a case life cycle \(on page 66\)](#)

## Step types

A step is the most granular element of your business process and represents a single action. Each step in a process has a type, based on the functionality that the step provides. By using a variety of step types, you can make your cases more interactive and ensure that your business processes include all relevant and necessary actions. For example, you can include steps that represent actions that users complete, such as providing information on a form, or automations that an application performs, such as sending automated notifications.

In a sample scenario, you can use steps in a Mortgage case type to retrieve a customer's financial history, prompt a case worker to calculate a new interest rate, and notify a case manager when the terms of the loan are ready for approval. You choose the step types when you define the case life cycle.

Pega Platform supports the following step types:



## Collect information

Assigns a task, which you define in a form with fields, to a user or work queue. For example, in a Loan request case type, you can add a step to collect users' personal details.

The following figure shows sample design-time and run-time presentation of a Collect information step:

The figure illustrates the design-time and run-time presentation of a Collect information step. On the left, the design-time view shows a task palette with the following items:

- Application review
- Collect information
- Collect personal details** (highlighted with a red box)
- Collect mailing address
- + More step

On the right, the run-time view shows a form titled "Adding a job candidate" with the following fields:

- Name
- Surname
- Phone number
- Street
- City
- Zip code

At the bottom of the form, there are two buttons: "Cancel" and "Create".

## Approve/Reject

Assigns a task to a user, typically a manager, to review case information and then decide whether to approve or reject the case. For example, in a case type to review job applications, you can add an Approve/Reject step so that a manager can approve a job candidate. You can provide additional configurations, such as enabling approval by email.



The following figure shows default design-time and run-time presentation of an Approve/Reject step with a text field for notes:



## Processes

Calls another process from the current process so that you can reuse an existing set of steps instead of repeatedly adding and configuring multiple steps. For example, you can reuse a process to collect feedback in many different case types.

## User actions

Assigns a predefined task to a user or work queue, such as adjusting a goal and deadline, or changing a stage. Resolving this step type requires a human action before a case can continue.

## Automations

Provides preconfigured functionality, such as sending an email or creating a case. Depending on the automation, you can configure additional options for a step. For example, for a step that automatically sends emails, you can apply business logic to define the recipient and provide the subject line and content. For more information about automations, see [Adding an automated step to a process \(on page 101\)](#).

## Form steps



Divides long forms into separate screens so that users can focus on more comprehensive units of information at once. For example, when a user completes a form to provide medical information, subsequent screens can display fields to capture their personal details, medical history, and current medical issues. Each set of information can be one screen, and one screen forms one step in the case life cycle.

The following figure shows a run-time presentation of two form steps:

The figure illustrates two overlapping form screens in a run-time presentation. The left screen, titled "Collect personal details", shows a progress indicator with a blue dot at the first step and a white dot at the second step. The right screen, titled "Collect medical information", shows a progress indicator with a white dot at the first step and a blue dot at the second step. The right screen contains several text input fields: "Describe your symptoms", "Current medications", "Allergies", and "Doctor that you want to contact". At the bottom of the right screen are "Cancel" and "Back" buttons. A "Continue" button is visible at the bottom of the left screen.

Form steps at run time

[Case life cycle elements \(on page 43\)](#)

## Adding single steps to processes

Model your business process with basic tasks that users or automations resolve, by adding steps to your case life cycle. When users complete steps, your case moves closer to its resolution and to achieving your business goal.



1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In a process, click **Step**, and then select a step type that you want to add.
  - To add a step that displays a form that users complete to provide data, click **Collect information**.
  - To add an approval step, click **Approve/Reject**.

Adding an approval step automatically creates an alternate stage in a case life cycle. For more information, see [Creating an alternate stage \(on page 58\)](#).

- To add a step that creates multiple screens with which users interact to provide data, click **Multi-step form**.

For more information, see [Adding a multistep form to a stage \(on page 70\)](#).

- To add an existing process as a single step, click **More > Processes**, and then select the process that you want to use.
- To add a step that requires human interaction, such as manually adjusting a service-level agreement, click **More > User action**, and then select an action that you want to use.
- To add an automation that an application performs, click **More > Automations**, and then select an automation that you want to use.

For more information, see [Adding an automated step to a process \(on page 101\)](#).

5. **Optional:** To provide a unique name for the step, in the text field, replace the default step label with a descriptive name.
6. Click **Save**.

---

[Adding an automated step to a process \(on page 101\)](#)

## Collecting information from a user

Gather the information that your business process requires by creating an assignment to collect that data from users. For example, a customer service representative can enter personal details and income information from a client during the loan request process.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.



3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, hover over a process in a stage, and then click **Step > Collect information**.
5. In the new text field, enter a unique name that describes the task.
6. In the properties panel, in the **Route to** list, identify the user or team that performs the task:
  - To assign the task to the user who last updated the case, select **Current user**.
  - To assign the task to another user in your application, select **Specific user**, and then configure the user parameters.

You can assign the task to a user by user name, a user by user reference, the reporting manager of the user who last updated the case, or a case participant.

- To assign the task to a team that shares a work queue, select **Work queue**, and then select a team.
- To assign the task to a worklist or a team that shares a work queue, select **Custom**, and then configure the team or worklist parameters.

For more information about configuring custom routing, see [Configuring custom routing logic for assignments \(on page 89\)](#).

- To automatically assign the task to a user with the appropriate availability, skill set, or workload, select **Use business logic**, and then click the **Configure business logic** icon.

For more information about configuring business-based routing, see [Assigning users automatically at run time with business logic \(on page 85\)](#).

7. **Optional:** In the **Set case status** field, set the status that the system assigns to the case when this process step starts:
  - To use an existing status, press the Down arrow key, and then select the status.
  - To create a new status, enter a unique name for the status.
8. **Optional:** To describe the purpose of the task, in the **Instructions for user** field, enter a description.
9. Click **Save**.

---

[Assignment shapes in processes \(on page 270\)](#)

[Case life cycle elements \(on page 43\)](#)

## Assigning tasks to users

Improve business process efficiency by assigning tasks to appropriate individuals or groups of individuals. For example, you can create a task for a CSR to follow up with a customer, and then route that assignment to the CSR work queue.



For relevant training materials, see the [Routing assignments to users](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, select a task that you want to route.
5. In the properties panel, in the **Route to** list, identify the user or team that performs the task:
  - To assign the task to the user who last updated the case, select **Current user**.
  - To assign the task to another user in your application, select **Specific user**, and then configure the user parameters.

You can assign the task to a user by user name, a user by user reference, the reporting manager of the user who last updated the case, or a case participant.

- To assign the task to a team that shares a work queue, select **Work queue**, and then select a team.
- To assign the task to a worklist or a team that shares a work queue, select **Custom**, and then configure the team or worklist parameters.

For more information about configuring custom routing, see [Configuring custom routing logic for assignments \(on page 89\)](#).

- To automatically assign the task to a user with the appropriate availability, skill set, or workload, select **Use business logic**, and then click the **Configure business logic** icon.

For more information about configuring business-based routing, see [Assigning users automatically at run time with business logic \(on page 85\)](#).

6. Click **Save**.

---

[Case life cycle elements \(on page 43\)](#)

[Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#)

[Adding optional actions to cases \(on page 354\)](#)

## Assigning users automatically at run time with business logic

Process your business cases with flexibility and efficiency by using business logic to automatically determine which individual or group of individuals receives an assignment in a case. Instead of entering specific names of users and teams, you can provide conditions that your application evaluates at run time to assign a task. For example, your application can evaluate which customer service representative (CSR) has the most appropriate skill set or workload to complete a step or task.



1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, click an assignment in a process.
5. In the **Step** section of the properties panel, click the **General** tab.
6. In the **Route to** list, select **Use business logic**, and then click the **Configure business logic** icon.
7. Select a routing method:

Actions	Steps
<b>Route the assignment to a specific user</b>	<ol style="list-style-type: none"> <li>a. In the <b>Route to</b> list, select <b>Operator</b>.</li> <li>b. In the <b>Value</b> field, press the Down arrow key, and then select the name of a user.</li> </ol>
<b>Route the assignment to a work queue</b>	<ol style="list-style-type: none"> <li>a. In the <b>Route to</b> list, select <b>Work queue</b>.</li> <li>b. In the <b>Value</b> field, press the Down arrow key, and then select the name of a work queue.</li> </ol>
<b>Route the assignment by using custom business logic</b>	<ol style="list-style-type: none"> <li>a. In the <b>Route to</b> list, select an option from the <b>Custom</b> section.</li> <li>b. Specify routing conditions by clicking <b>Edit parameters</b> and completing the <b>Edit parameters</b> window. You can route the assignment based on availability and skill set, or based on availability, skill set, and workload.</li> </ol>

8. Define the routing condition for the assignment:
  - a. In the list of values to evaluate, select the name of a field in your data model or a when condition.
  - b. In the comparator list, select a comparator.
  - c. In the value field, enter or select a value that your application compares with the field in the data model or the when condition.
  - d. **Optional:** To define more conditions that apply to this routing method, click the **Add row** icon, and then repeat steps [8.a \(on page 86\)](#) through [8.c \(on page 86\)](#)

- e. If you add multiple conditions, between the rows, select the **and** or **or** operator to define how to evaluate the conditions.

If you select **and**, the condition evaluates to true when all of the rows evaluate to true.  
If you select **or**, the condition evaluates to true if at least one of the rows evaluates to true.

9. **Optional:** To support complex business logic, click **Add condition**, and then repeat steps [7 \(on page 86\)](#) and [8 \(on page 86\)](#).

At run time, your application selects the first routing method that meets the conditions that you define.

10. **Optional:** To change the order in which your application evaluates the conditions, define a new order by dragging the conditions.
11. In the **otherwise** section, define a routing method that you want to use when no conditions return a true value by repeating step [7 \(on page 86\)](#).
12. Click **Submit**.

At run time, an application evaluates the conditions and automatically routes an assignment to the correct user or work queue. If an application evaluates all conditions as false, it routes an assignment by using an alternative method that you provide in the **otherwise** section.

[Assignment shapes in processes \(on page 270\)](#)

[Configuring custom routing logic for assignments \(on page 89\)](#)

## Configuring custom business logic-based routing

To process cases quicker, ensure that assignments are routed to the most appropriate workers by using custom APIs for business logic-based routing. You can choose from default APIs or add custom APIs to meet your unique business needs. You can also override APIs to modify lists of available operators and work queues, so that the lists contain only relevant workers. For example, in a field service application you create an assignment with business logic that routes the service request to the operator whose postal code matches the location of the customer who raised the request. At run time, the system compares customer's postal code with the conditions configured in the business logic. The system then routes the assignment to the operator whose postal code meets the condition.

You configure the business logic-based routing APIs by completing the following tasks:

## Related information

[Assigning users automatically at run time with business logic \(on page 85\)](#)

## Adding custom routing options in business logic-based routing

Process cases quicker by creating custom routing options for assignments. To ensure that assignments are routed to the most appropriate worker, edit the parameters of your custom options.

In the business logic-based routing, you can use the following default custom options to route assignments:

- Route based on availability and skill set
- Route based on availability, skill set, and work load

In a dialog box, you can enter values for the parameters that are needed for the selected option.

To meet your specific business needs, create your own custom options by overriding the *pyGetRoutingAPIs* data transform:

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Data Model** category, and then click **Data Transform**.
3. On the **Instances of Data Transform** tab, open the *pyGetRoutingAPIs* data transform.
4. On the **Definition** tab, set the *pyStandardValue* parameter as equal to the name of your API.
5. Set the *pyPromptValue* parameter as equal to the label of your API.
6. Configure the return of the API by setting the *pyAPIReturnType* parameter:
  - To route the assignment to a worker, enter "OPERATOR".
  - To route the assignment to a work queue, enter "WORKQUEUE".
7. Click **Save as**.
8. Click **Create and open**.
9. **Optional:** Customize the source of properties that are mapped with parameters:
  - a. In the navigation pane of Dev Studio, click **Records**.
  - b. Expand the **Technical** category, and then click **Activity**.
  - c. On the **Instances of Activity** tab, open *pySetRuleKeys*.
  - d. Set the *Param.RuleClass* parameter as equal to the class that you want to use as the source for the parameters of your custom API.
  - e. Click **Save as**.
  - f. Click **Create and open**.
10. **Optional:** Include custom UI elements in the **Parameters** section:
  - a. In the navigation pane of Dev Studio, click **Records**.
  - b. Expand the **User Interface** category, and then click **Section**.



- c. On the **Instances of Section** tab, open *pyRuleParamsUIExtension* of a class *Data-MO-Activity-Router*.
- d. Customize the section to meet your specific needs.
- e. Click **Save as**.
- f. Click **Create and open**.

[Assigning users automatically at run time with business logic \(on page 85\)](#)

## Modifying lists of operators and work queues for business logic-based routing assignments

Modify the list of operators and the list of work queues that are available for business logic-based routing of assignments so that the lists contain only relevant workers. Modify the lists to ensure that the assignment is routed to an appropriate person or work queue. For example, you can filter the list of operators, so that you can only select workers with specific privileges.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Technical** category, and then click **Activity**.
3. Open *pyGetOperatorsOrWorkQueues*.
4. Modify the activity to meet your specific needs, for example, add a step to view the list of operators that have certain privileges:
  - a. Click **Add a step**.
  - b. Enter a label for the step.
  - c. Click **When**, and then select **Enable conditions before this action**.
  - d. Define when criteria. For more information, see *Activity form - Completing the Steps tab - Entering preconditions (on page 85)*.
  - e. Click **Submit**.
5. Click **Save as**.
6. Click **Create and open**.

### Related information

[Assigning users automatically at run time with business logic \(on page 85\)](#)

## Configuring custom routing logic for assignments

Determine which user or team receives an assignment in a case by configuring custom routing logic. By using an activity instead of business logic, you can access advanced features that support complex use cases.



As a best practice, use business logic instead of an activity to route assignments, because this method is more flexible and easier to maintain. For more information, see [Assigning users automatically at run time with business logic \(on page 85\)](#).

1. In the navigation pane of Dev Studio, click **App**.
2. On the **Classes** tab, expand the case type where you want to edit the assignment, and then click **Process > Flow**.
3. In the **Flow** form, on the **Diagram** tab, double-click an assignment.
4. In the **Assignment properties** dialog box, in the **Routing** section, from the **Route to** list, select **Custom**.
5. In the **Assignment type** list, choose a type of destination for the assignment:
  - To route the assignment to a specific user or worklist, select **Worklist**.
  - To route the assignment to a team or work queue, select **Work queue**.
6. Specify the router activity that sets the `AssignTo` parameter to the name of a worklist or a work queue:
  - To use an existing activity, in the **Router** field, press the Down arrow key, and then select the router activity.
  - If you do not have an activity defined, in the **Route to** list, select **Operator**, and then provide the name of the property that stores an operator ID.
7. If the router activity accepts parameters, in the **Parameters** section, enter values for these fields.
8. **Optional:** If you route the assignment to a work queue, to send the assignment to the user who is currently processing the case when no work queue is found, in the **Advanced** section, select the **Use current operator when work queue not found**.
9. Click **Submit**.
10. Click **Save**.

---

[Assignment shapes in processes \(on page 270\)](#)

## Adding instructions to assignments

Provide users with guidance on how to complete their tasks by adding instructions to assignments. When you add instructions for completing steps, you give users independence, and accelerate case resolution.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, click the step that you want to supplement with instructions, for example, **Collect information**.
4. In the **Step** property panel, click the **General** tab.



5. In the **Instructions for user** field, enter additional information about the step.
6. Click **Save**.

At run time, the instructions appear at the top of the form that the user completes, as shown in the following

figure:

**KM Collect information**  
Enter personal details:

Name

Surname

*Instructions for an assignment*

[Step types \(on page 79\)](#)

[Collecting information from a user \(on page 83\)](#)

[Processes in a case life cycle \(on page 66\)](#)

## Requesting approval from users

Ensure that your business processes meet your organization policies and requirements by requesting approval from users. After reviewing case details, users can approve or reject a case and bring the case closer to resolution.

For example, a manager can approve or reject a job candidate in a hiring process after reviewing the candidate's CV and conducting the job interview. By configuring an approval step, you ensure that only the appropriate person, such as a manager, can make certain decisions in a case.

For relevant training materials, see the [Designing an approval process](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, in a stage where you want to add an approval step, click **Step > Approve/Reject**.

Your case type includes an *Approval Rejection* alternate stage.

5. In the text field that appears, enter a unique name that describes the step.
6. On the **General** tab of the **Step** properties pane, route the approval step:

- To route the approval step to a user in your application, in the **Route to** list, select **Specific user**, and then enter the user details.

You can assign the task to a user by user name, user reference, the reporting manager of the user who last updated the case, or to a case participant.



**Note:** You can route the approval step only to participants that have an account in your application.

- To route the approval step to a team that shares a work queue, in the **Route to** list, select **Work queue**, and then select or enter a team.
- To route the approval step to a user that your application assigns at run time based on conditions that you define, select **Use business logic**, and then click **Configure business logic** to define the conditions.

For example, an application can route an assignment to a user with the most appropriate skill set and availability. For more information, see [Assigning users automatically at run time with business logic \(on page 85\)](#).

7. In the **Step** properties pane, click the **Flow** tab, and then define what happens when a user approves the case:

Choices	Actions
<p><b>Approve and move case to the next step</b></p>	<p>a. In the <b>If approved then</b> list, select <b>Continue</b>.</p> <p>b. <b>Optional:</b> To change the case status, in the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user approves the case.</p>
<p><b>Approve and change stage</b></p>	<p>a. In the <b>If approved then</b> list, select <b>Change stage</b>.</p> <p>b. In the <b>To</b> list, select the stage that you want to assign to the case when a user gives approval.</p> <p>c. <b>Optional:</b> To change the case status, in the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user approves the case.</p>

8. Define what happens when a user rejects the case:

Choices	Steps
<b>Reject and move case to the next step</b>	<ol style="list-style-type: none"> <li>In the <b>If rejected then</b> list, select <b>Continue</b>.</li> <li>In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ol>
<b>Reject and change stage</b>	<ol style="list-style-type: none"> <li>In the <b>If rejected then</b> list, select <b>Change stage</b>.</li> <li>In the <b>To</b> list, select the stage that you want to assign to the case when a user rejects the case.</li> <li>In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ol>
<b>Reject and resolve</b>	<ol style="list-style-type: none"> <li>In the <b>If rejected then</b> list, select <b>Resolve</b>.</li> <li>In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ol>

9. Click **Save**.

[Requesting approval by email and push notification \(on page 93\)](#)

[Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#)

[Creating views for case types \(on page 173\)](#)

[Case life cycle elements \(on page 43\)](#)

## Requesting approval by email and push notification

Provide greater flexibility and convenience for users of your application when they process a case, by enabling approval by email and push notification. Users can then accept, reject, or take other actions on a case without having to log in to the application.

The following process describes an example of a case approval by email:

- Case workers that process a case in an end user portal move the case through the different steps in the life cycle.
- The case reaches an approval step in the life cycle that is configured to send an email with an approval request.



- Your application sends the email to the reporting manager of the case worker that last updated the case. The email includes an embedded prompt to accept or reject the case step.
- The manager clicks a button in the email message, which generates an email response that includes their decision and sends the email back to the application.
- Your application receives the response, and either moves the case forward when the manager decides to approve the case, or moves the case to an alternate stage in the life cycle when the manager decides to reject the case.

You can also configure case behavior after an approval or a rejection to meet your specific business needs. For more information, see [Requesting approval from users \(on page 91\)](#).

You can send personalized emails by including properties in messages. Your application then autopopulates the properties at run time with data corresponding to the relevant case. For example, you can include a property that corresponds with a case ID or a recipient name. After you create email content, you can save the message in a template library, so that you can reuse the message in the future. You can also include attachments to provide more information and context before a user accepts or rejects a case. To reach wider audiences and provide even more flexibility, you can also send approval requests to users of mobile apps by enabling push notifications in your app. The subject that you define for the email is also a push notification text.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Workflow** tab, click **Life cycle**.
3. Click the Approve/Reject step for which you want to add email or push notification approval requests.
4. In the **Step** properties pane, in the **Enable approval from** section, select how you want to send the approval message:
  - To request approval by email, select the **Email** check box.
  - To request approval by push notification, select the **Mobile** check box.

You can select both email and push notification forms.

5. In the **Email subject / Push notification message** field, enter text that you want to use as an email subject or a push notification message.

You can reference property names in the subject to make it more dynamic and meaningful.

At run time, the application autopopulates the properties with relevant data. For example, at run time the email subject is `Case LoanRequest-123 is waiting for your approval`.

6. If you request approval by email, define your message content:

Choices	Actions
<p><b>Compose a message</b></p>	<ol style="list-style-type: none"> <li>In the <b>Email content</b> list, select <b>Custom</b>, and then click <b>Compose email content</b>.</li> <li>In the <b>Compose message</b> window, create your message by using the rich text editor.</li> <li><b>Optional:</b> To add the email message to a template library to reuse in the future, click <b>Actions &gt; Add to approval template library</b>, in the <b>Template name</b> field provide the template name, and then click <b>Save</b>.</li> <li>Click <b>Done</b>.</li> </ol>
<p><b>Reuse a template</b></p>	<ol style="list-style-type: none"> <li>In the <b>Email content</b> list, select <b>Use existing</b>.</li> <li>In the list of email templates, select a template that you want to use. You can select templates from your template library as well as from the default templates.</li> <li><b>Optional:</b> To modify the message to meet your unique business needs, click <b>Compose email content</b>, modify the message by using a rich text editor, and then click <b>Done</b>.</li> </ol>

7. **Optional:** To add attachments to the email, select the **Include attachments** check box, and then select the files that you want to upload:

Actions	Choices
<p><b>Include all case attachments</b></p>	<p>Select <b>All case attachments</b>.</p>
<p><b>Include only specific case attachments</b></p>	<ol style="list-style-type: none"> <li>Select <b>Choose attachments</b>.</li> <li>Click <b>Add attachment</b>.</li> </ol>

Actions	Choices
	<p>c. In the drop-down list, select whether you want to include an attachment of a particular category, or add the content of a field in your case.</p> <p>d. In the field below, select an attachment category or a field that you want to include as an attachment.</p> <p>e. To add more attachments, repeat steps <a href="#">7.b (on page 95)</a> through <a href="#">7.d (on page 96)</a>.</p>

8. Click **Save**.

When the case life cycle reaches the approval step, your application sends an email or a push notification, based on the approval step configuration. A user who receives the approval request can take action on the case without logging in to the application.

[Requesting approval from users \(on page 91\)](#)

[Case life cycle elements \(on page 43\)](#)

## Configuring cascading approvals

Receive approval from people on different levels of your organizational chart by configuring a system of cascading approval. For example, you can create a case that requires authorization first from the reporting manager of the worker processing the case, and then from the senior manager.

For relevant training materials, see the [Cascading approvals](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, click the approval step that you want to edit.
3. In the **Step** properties panel, on the **General** tab, in the **Approval flow type** list, select **Cascading**.
4. In the **Approval based on** list, select **Reporting structure**.
5. In the **Approval to be completed by** list, specify the user who is required to give their approval:



- To assign the approval to the reporting manager of the user who processes the case, select **Reporting manager**.
- To assign the approval to a manager of the work group in which the user who processes the case belongs, select **Workgroup manager**.

6. Specify how many levels of approval a case requires:

Choices	Actions
<b>One level of approval</b>	Expand the <b>Approval level</b> section, and then select <b>One</b> .
<b>Request approval from managers of all levels in the organizational chart</b>	Expand the <b>Approval level</b> section, and then select <b>All</b> .
<b>Create custom approval logic</b>	<ol style="list-style-type: none"> <li>Expand the <b>Approval level</b> section, and then select <b>Custom</b>.</li> <li>Click <b>Update custom levels</b>.</li> <li>In the window that appears, click <b>Add custom approval</b>.</li> <li>In the <b>When</b> field, enter a when condition. The when condition returns a Boolean expression that determines the manager required to approve the case.</li> <li>In the <b>Levels of approval</b> field, enter how many levels of approval a case requires.</li> <li><b>Optional:</b> To add more conditions, repeat steps <a href="#">6.c (on page 97)</a> through <a href="#">6.e (on page 97)</a>.</li> <li>Click <b>Submit</b>.</li> </ol>

7. In the **Step** properties panel, click the **Flow** tab, and then define what happens when a user approves the case:

Choices	Actions
<b>Approve and move case to the next step</b>	<ul style="list-style-type: none"> <li>a. In the <b>If APPROVED then</b> list, select <b>Continue</b>.</li> <li>b. <b>Optional:</b> To change the case status, in the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user gives their approval.</li> </ul>
<b>Approve and change stage</b>	<ul style="list-style-type: none"> <li>a. In the <b>If APPROVED then</b> list, select <b>Change stage</b>.</li> <li>b. In the <b>To</b> list, select the stage that you want to assign to the case when a user gives their approval.</li> <li>c. <b>Optional:</b> To change the case status, in the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user gives their approval.</li> </ul>

8. Define what happens when a user rejects the case:

Choices	Steps
<b>Reject and move case to the next step</b>	<ul style="list-style-type: none"> <li>a. In the <b>If REJECTED then</b> list, select <b>Continue</b>.</li> <li>b. In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ul>
<b>Reject and change stage</b>	<ul style="list-style-type: none"> <li>a. In the <b>If REJECTED then</b> list, select <b>Change stage</b>.</li> <li>b. In the <b>To</b> list, select the stage that you want to assign to the case when a user rejects the case.</li> <li>c. In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ul>

Choices	Steps
<b>Reject and resolve</b>	<ol style="list-style-type: none"> <li>a. In the <b>If REJECTED then</b> list, select <b>Resolve</b>.</li> <li>b. In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ol>

9. Click **Save**.

[Completing work on time \(on page 324\)](#)

[Sending event notifications from cases \(on page 140\)](#)

## Configuring approvals with an authority matrix

Create a system for obtaining approval from people in different parts or departments of your organization, by creating an authority matrix. For example, you can create a case that processes business trip expenses, and then configure an authority matrix to request approvals from a reporting manager and a financial worker.

For relevant training materials, see the [Cascading approvals](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, click the approval step that you want to edit.
3. In the **Step** properties panel, on the **General** tab, in the **Approval flow type** list, select **Cascading**.
4. In the **Approval based on** list, select **Authority matrix**.
5. Define the users required to give their approval in the step:
  - a. **Optional:** In the **Decision table for matrix** field, enter a decision table that determines who is required to give their approval in the step.  
Decision table populates the page list property. Number of the entries in the page list property defines the number of users required to give their approval. The approver property that you include in the decision table, holds the approver information in the page list property.
  - b. In the **Page list property** field, enter a page list that stores the users giving their approval.
  - c. In the **Approver property** field, enter a property that references the approving users.
6. In the **Step** properties panel, click the **Flow** tab, and then define what happens when a user approves the case:

Choices	Actions
<b>Approve and move case to the next step</b>	<ul style="list-style-type: none"> <li>a. In the <b>If APPROVED then</b> list, select <b>Continue</b>.</li> <li>b. <b>Optional:</b> To change the case status, in the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user gives their approval.</li> </ul>
<b>Approve and change stage</b>	<ul style="list-style-type: none"> <li>a. In the <b>If APPROVED then</b> list, select <b>Change stage</b>.</li> <li>b. In the <b>To</b> list, select the stage that you want to assign to the case when a user gives their approval.</li> <li>c. <b>Optional:</b> To change the case status, in the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user gives their approval.</li> </ul>

7. Define what happens when a user rejects the case:

Choices	Steps
<b>Reject and move case to the next step</b>	<ul style="list-style-type: none"> <li>a. In the <b>If REJECTED then</b> list, select <b>Continue</b>.</li> <li>b. In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ul>
<b>Reject and change stage</b>	<ul style="list-style-type: none"> <li>a. In the <b>If REJECTED then</b> list, select <b>Change stage</b>.</li> <li>b. In the <b>To</b> list, select the stage that you want to assign to the case when a user rejects the case.</li> <li>c. In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ul>

Choices	Steps
<b>Reject and resolve</b>	<ol style="list-style-type: none"> <li>In the <b>If REJECTED then</b> list, select <b>Resolve</b>.</li> <li>In the <b>Set status</b> field, select or enter the status that you want to assign to the case when a user rejects the case.</li> </ol>

8. Click **Save**.

[Completing work on time \(on page 324\)](#)

[Sending event notifications from cases \(on page 140\)](#)

## Adding an automated step to a process

Save time and design more efficient business processes by adding preconfigured steps to your case types. When you include a preconfigured step in your case life cycle, you provide an automation that an application performs to bring a case closer to a resolution. For example, you can include a step that automatically sends a notification to a manager when a case awaits approval.

For greater flexibility and efficiency, Pega Platform offers a wide variety of automations that you can incorporate into your case life cycle. To ensure that your application meets all of your business requirements, each step provides additional configuration. For example, for a preconfigured step that sends a notification, you can configure how your application determines the recipient and define the notification content, as shown in the following video:

[https://players.brightcove.net/1519050010001/default\\_default/index.html?videoId=6275489038001](https://players.brightcove.net/1519050010001/default_default/index.html?videoId=6275489038001)

You can add the following automations to your case life cycle:

[Case life cycle elements \(on page 43\)](#)

[Creating a Microjourney for customer success \(on page \)](#)

[Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#)

[Defining an escalation action for an incomplete assignment \(on page 337\)](#)

## Attaching content to a case

Facilitate business interactions and provide users with the supporting information for a case by attaching relevant correspondence and documentation to a case type. Use the Attach Content shape in design time to control the kind of information that case workers add to a case.

For example, in a loan request case, you can attach a certificate of employment form, which the case worker can provide to a customer to ensure that customers always complete the standard form.



For relevant training materials, see the [Organizing and managing access to case attachments](#) module on Pega Academy.

1. Add the Attach Content shape to the life cycle of your case type:
  - a. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the stage in which you want users to be able to attach content, click **Step > More > Automations > Attach content**.
  - c. Click **Select**.
2. In the **Step** properties pane, select and configure the attachment type:

Choices	Actions
<p><b>Attach a document</b></p>	<ol style="list-style-type: none"> <li>a. Click <b>Attach a file</b>.</li> <li>b. Click <b>Upload</b>.</li> <li>c. In the <b>Upload a file</b> window, click <b>Choose File</b>, and then select the file to upload.</li> <li>d. Click <b>Done</b>.</li> <li>e. In the <b>Select file</b> list, select the file to attach.</li> <li>f. In the <b>Attachment name</b> field, enter a descriptive name for the file.</li> <li>g. <b>Optional:</b> To change the business classification of the attachment, select an option in the <b>Attachment category</b> field.</li> </ol> <div data-bbox="938 1549 1468 1848" style="border: 1px solid #0070C0; padding: 10px; margin-top: 20px;"> <p><b>Note:</b> You can add attachment categories, which group attachments according to their business classification, on the <b>Settings</b> tab, in the <b>Attachment categories</b> section.</p> </div>

Choices	Actions
	<p data-bbox="941 388 990 441">i</p> <p data-bbox="1015 220 1437 556">For example, you can add the Proof of identity and Proof of address attachment categories that group two different sets of documents required in processing a request for a service offered by the government.</p>
<b>Attach a URL</b>	<ol data-bbox="901 640 1461 1165" style="list-style-type: none"><li>Click <b>Attach a link</b>.</li><li>In the <b>URL</b> field, enter a URL that points to an external file or resource on the Internet.</li><li>In the <b>URL name</b> field, enter a descriptive name for the link.</li><li><b>Optional:</b> To change the business classification of the attachment, select an option in the <b>Attachment category</b> field.</li></ol> <p data-bbox="941 1554 990 1606">i</p> <p data-bbox="1015 1249 1437 1533"><b>Note:</b> You can add attachment categories, which group attachments according to their business classification, in the <b>Attachment categories</b> section, on the <b>Settings</b> tab.</p> <p data-bbox="1015 1564 1437 1900">For example, you can add the Proof of identity and Proof of address attachment categories that group two different sets of documents required in processing a request for a service offered by the government.</p>

Choices	Actions
<p><b>Attach a note (available only in Dev Studio)</b></p> <p><b>Note:</b> This option is for backward compatibility only and is not recommended. Instead, use Pulse to add notes and collaborate on cases.</p> <p><b>For more information, see <a href="#">Configuring Pulse for case types (on page 220)</a>.</b></p>	<ol style="list-style-type: none"> <li>In the navigation pane of Dev Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>Perform steps <a href="#">1b (on page 102)</a> through <a href="#">1c (on page 102)</a>.</li> <li>In the <b>Step</b> properties pane, click <b>Attach A Note</b>.</li> <li>In the <b>Note</b> text field, provide a note that you want to attach.</li> <li>In the <b>Attachment description</b> field, enter a descriptive name for the note.</li> </ol>

- Optional:** To provide more information for auditing purposes, in the **Audit note** field, enter additional information.
- Click **Save**.

At run time, users can open and download an attachment in a case.

[Controlling access to case attachments \(on page 457\)](#)

[Flow shapes \(on page 251\)](#)

[Adding an automated step to a process \(on page 101\)](#)

## Attaching screen captures to a case

To document values that users enter during case processing, create screen captures of user inputs, and then save the captures as PDF files. By documenting input values, you can understand how users interact with a case and, after case resolution, analyze the information that users provide. For example, you can create a PDF that contains details of a complaint that a customer files so that you can download the PDF for additional processing. For example, you can then download the PDF, print it, and attach to relevant offline documentation; or download the PDF and send via email to interested stakeholders.

**Note:**





- PDF is a restrictive format that forces hard constraints on settings such as margins or layouts. Consequently, the *HTMLToPDF* rule does not support the conversion of dynamic layout groups. As a best practice, use free form layouts or smart layouts with forms that you want to print to PDF.
- Only standard Pega Platform applications support attaching PDF documents to cases. Applications that you build on Cosmos React do not support this feature.

1. Add the Create PDF shape to the case life cycle:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - c. In a process in which you want to add the shape, click **Step > More > Automations > Create PDF**.
  - d. Click **Select**.
2. In the **Step** properties pane, in the **Select view** list, select a view that you want to include as a PDF screen capture.
3. In the **Step** properties pane, in the **Section name** field, select a section that you want to include as a PDF screen capture.
4. In the **Description** field, enter a short description of the attachment.
5. **Optional:** To describe the business classification of the screen capture, in the **Attachment category** field, select an option.
6. In the **PDF orientation** section, specify the file layout.
7. In the **PDF name** field, enter a descriptive name for the document, for example, `Complaint`.
8. In the **Attachment category** list, select an option that describes the business classification of the screen capture.
9. **Optional:** In the **Audit note** field, enter the audit note.
10. **Optional:** To provide a link to this step from bread crumb trail navigation, select **Enable navigation link**.
11. Click **Save**.

At run time, your application takes a screen capture of the section that you select, and then saves the capture as a PDF file in the **Recent content** section of a case.

[Attachment types \(on page 458\)](#)

[Restricting user actions for case attachments \(on page 459\)](#)



[Flow shapes \(on page 251\)](#)

[Adding an automated step to a process \(on page 101\)](#)

## Calling a data transform from a case

Save time and speed up case processing by prepopulating data in your case. When you add the Run data transform shape to a case life cycle, your application converts data from an external source to a format that fits your case. For example, while processing a purchase order, you can populate the shipping address with the data that the user provides as their billing address.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the process in which you want to call a data transform, click **Step > More > Automations > Run data transform**, and then click **Select**
3. In the **Step** properties pane, in the **Run data transform** list, select the data transform that you want to call in this step.
4. **Optional:** In the **Audit note** field, enter some helpful extra text.
5. Click **Save**.

---

[Referencing properties \(on page \)](#)

[Data transforms \(on page \)](#)

## Creating contextual cases

To provide additional processing in your business processes, create a contextual case that starts when your current case is in progress. Creating contextual cases helps you deliver more granular and flexible applications. Instead of defining one long, complex case type, you can trigger a contextual case at an appropriate point in your main case type. For example, in an application to review insurance claims after car accidents, you can start a contextual case to collect and review materials so that a customer service representative (CSR) can estimate the damage before returning to the main case.

Creating contextual cases in Dev Studio is suitable for advanced developers. To take advantage of low-code solutions in App Studio, see [Creating a child case type \(on page 31\)](#).

1. Add the Create case shape to the life cycle of your case:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the process in which you want to add the Create case shape, click **Step > More > Automations > Create case**.



- c. Click **Select**.
  - d. In the **Create case** field, enter the name of the child case type, for example *Damage estimates*.
2. In the **Step** properties pane, define the case type that you want to create:
- To start an independent case type, select **Create case**.
  - To start a child case of the current case, select **Create child case(s)**.

If you create a child case type, you can only complete your main case type after the child case type reaches resolution.

3. Select a case type to start:

Choices	Actions
<b>Create a new case type</b>	<ol style="list-style-type: none"> <li>a. In the <b>Create the following case</b> list, select <b>Create new Case type</b>.</li> <li>b. In the <b>Case type name</b> field, enter the name of your case type.</li> <li>c. Click <b>Create</b>.</li> </ol>
<b>Use an existing case type</b>	In the <b>Create the following case</b> list, select the case type that you want to use.
<b>Use an existing case type by selecting a class</b>	<ol style="list-style-type: none"> <li>a. In the <b>Create the following case</b> list, select <b>Other</b>.</li> <li>b. In the <b>Class</b> field, press the Down arrow key, and then select the class that stores the case type that you want to use. When you select a class, you do not need to provide a run-time case type name.</li> </ol>

4. **Optional:** If you create a child case type, to create more than one case type, select the **Create multiple cases using a list** check box, and then, in the **List field** field, press the Down arrow key, and then select a field group that determines the number of cases to create.
5. **Optional:** To propagate data from your current case to the new case, in the **Data transform** field, press the Down arrow key, and then select the name of a data transform.
6. **Optional:** If you create a new case type, to obtain a reference to the new case that you can use in the current case, in the **Property to store ID of case** field, press the Down arrow key, and then select the name of a single-value property.

7. **Optional:** To provide more information for auditing reasons, in the **Audit note** field, enter the audit note.
8. Click **Save**.

[Case types \(on page 28\)](#)

[Steps in a case life cycle \(on page 78\)](#)

## Collecting digital signatures in a case

Enhance your business transactions with relevant information by electronically signing, reviewing, and sharing documents. When you integrate your case types with DocuSign, you can collect digital signatures for your case attachments, and as a result, automate the review process for documents and resolve cases more quickly. When you collect digital signatures, you avoid producing and exchanging printed documents, which might take longer. For example, in a hiring process, a hiring manager might digitally sign a decision to approve a job candidate.

1. Add the Send via DocuSign shape to the life cycle of your case:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the process in which you want to add the Send via DocuSign smart shape, click **Step > More > Automations > Send via DocuSign**.
  - c. Click **Select**.
2. In the **Step** properties pane, in the **Envelope** field, enter the envelope that you want to use. A DocuSign envelope stores information that is necessary to make a successful signature request to DocuSign.
3. **Optional:** To provide more information about this step for auditing purposes, in the **Audit note** field, enter the information that you want to leave.
4. **Optional:** To allow users to navigate to this step from the bread crumb navigation at run time, select the **Enable navigation link** check box, and then specify additional options:
  - a. **Optional:** To allow users to navigate to the step only after this step is complete, select the **Only allow navigating back to this step** check box.
  - b. **Optional:** To enable post-processing or validation when users navigate back from the step, select the **Perform post-processing when navigating away from step** check box.
5. Click **Save**.

### Related information

Configuring a signature request to DocuSign (on page )

[Attaching content to a case \(on page 101\)](#)



## Finding duplicate cases

Save time by identifying potential duplicate cases in your application and ensuring that each case represents a unique request. Potential duplicate cases match a set of required and weighted conditions. For example, you can check if a customer creates a loan request for a specific amount of money before a previous loan request from the same customer reaches resolution.

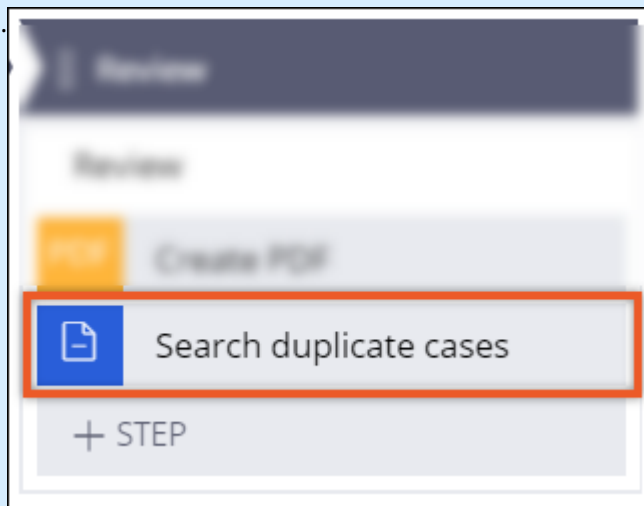
[Adding an automated step to a process \(on page 101\)](#)

## Searching duplicate cases

Save time and minimize creation of similar cases by adding a Search duplicate cases shape to your case life cycle.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, on the **Workflow** tab, click **Life cycle**.
4. In a process to which you want to add a Search duplicate cases shape, click **Step > More > Automations > Search duplicate cases**.
5. Click **Select**.

Your case type now includes a Search duplicate cases shape, as shown in the following figure.



Search duplicate cases shape

#unique\_143 (on page )

[Identifying duplicate cases](#)

## Defining basic conditions for case types

Improve the performance of your duplicate case search and limit the number of potential duplicates by defining basic conditions for potential duplicate cases.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the case working area, on the **Workflow** tab, click **Life cycle**.
3. In the **Case life cycle** section, click the **Search duplicate cases** step for which you want to define basic conditions.
4. In the step property pane, click **Add basic condition**.
5. In the dialog box that appears, in the **Potential duplicate**, **Comparator**, and **Current case** fields, enter the property names and values that you want to use to determine duplicates in a comparison against the existing cases.

The application uses these values to compare existing cases against the current case to identify potential duplicates.

**Note:** You need to expose the properties that are referenced by an exact condition as columns in the database. For more information, see [Optimizing database properties \(on page 110\)](#).

6. **Optional:** To define more basic conditions, in the dialog box, click **Add a row**, and then go to step 5 ([on page 110](#)).
7. **Optional:** To mark the case as a potential duplicate only when the case meets all basic conditions, in the field below **When**, select **and**.
8. **Optional:** To mark the case as a potential duplicate when the case meets at least one basic condition, in the field below **When**, select **or**.
9. Click **OK**.
10. Click **Save**.

At run time, the application evaluates the basic conditions first to limit the list of potential duplicates. If the case meets the basic conditions, the system analyzes the weighted conditions.

The results are displayed in tabular form. You can decide whether to close a case as a duplicate or ignore the potential duplicate and continue case processing.

In the results table, you can preview the matching property values of a potential duplicate case and the total score contribution number by clicking **Why is this shown here?**. A dialog box is displayed

that provides the reason for marking the case as a potential duplicate, which helps you make an informed decision about what action to take.



**Note:** Only simple match scores are shown in the dialog box. The *Date time* and *Date* property types are complex match scores and do not display the values. However, the total score contribution of complex matches is shown in the **Other conditions** section at the bottom of the dialog box, so the lack of display has no impact on duplicate case evaluation.

Restrictions on properties used in duplicate tracking (on page )  
[Adding an automated step to a process \(on page 101\)](#)

## Defining a threshold with weighted conditions for a case type

Improve the performance of your duplicate case search and limit the number of potential duplicates by defining weighted conditions for a potential duplicate case. Determine whether the case is a potential duplicate by assigning a relative weight to each condition, and by calculating the total weight of a case.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the case working area, on the **Workflow** tab, click **Life cycle**.
3. In the **Case life cycle** section, click the **Search duplicate cases** step for which you want to define the weighted conditions.
4. In the step property panel, click **Add weighted condition**.
5. In the dialog box that appears, in the **Weight** field, enter a value between 1 and 100 that you want to add to the total weight of a case when the case meets the condition.




**Note:** If you want the application to display additional information in the notification about duplicate cases, include the properties that contain that information as columns in a version of the *pyCaseMatchProcessingTemplate* report definition that applies to the class of your case type. For more information, see [Optimizing database properties \(on page \)](#).

6. In the **Potential duplicate**, **Comparator**, and **Current case** fields, enter the property names and values that you want to use to determine duplicates in a comparison against the existing cases.

The application uses these values to compare existing cases against the current case to identify potential duplicates.

7. Click **OK**.
8. **Optional:** To add more weighted conditions, repeat steps 4 ([on page 111](#)) through 7 ([on page 112](#)).
9. In the **And weighted conditions sum at least** field, enter a sum of weights that represents a threshold when the case is a potential duplicate.

 **Tip:** The threshold value should be lower than the sum of all weighted conditions.

10. Click **Save**.

At run time, the application evaluates the conditions that have different weights to determine which cases are likely to be duplicates. Users can decide whether to close a case as a duplicate or ignore the potential duplicate and continue. A dialog box provides the reason for marking the case as a potential duplicate, which helps users make an informed decision.



**Note:** Only simple match scores are shown in the dialog box. The *Date time* and *Date* property types are complex match scores and do not display the values. However, the total score contribution of complex matches is shown in the **Other conditions** section at the bottom of the dialog box, so the lack of display has no impact on duplicate case evaluation.

---

Restrictions on properties used in duplicate tracking ([on page 100](#))  
[Adding an automated step to a process \(on page 101\)](#)

## Generating case documents

To provide your stakeholders with crucial information about your business processes in an automated and convenient way, generate case documents as part of your case life cycle. You create case documents by using templates to ensure that you deliver relevant, required data.

When you create templates for your case documents, you include tags that your application populates with case data during case processing. You can include tags for the following entities:



- Fields that a case data model contains
- Attachments
- Views
- Correspondence

For example, you can add tags for the case ID, goal and deadline, or case participants. You can also reuse one template for different cases, because every time your application provides information that is specific to the current case. You create document templates in a text editor and save files in the .docx format.



**Note:** Only standard Pega Platform applications support generating case documents. Applications that you build on Cosmos React do not support this feature.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In a process to which you want to add a **Generate document** shape, click **Step > More > Automations > Generate document**.
4. In the **Step** properties panel, select a document to use as a template:

#### Choices

##### Create a new template

#### Actions

- a. Create a new document in a .docx format in a word processor.  
The case document that your application generates reflects the styling that you use when you create a template, for example, the font and size.
- b. In the **Step** properties panel, click **View and use fields**.
- c. In the **View and use fields window**, search for a tag that you want to include in your document, and then click **Copy tag**.
- d. Paste the tag where you want to include information in your document.

Choices	Actions
	<ul style="list-style-type: none"> <li>e. Repeat steps <a href="#">4.c (on page 113)</a> and <a href="#">4.d (on page 113)</a> for the tags that you need, and then save the document.</li> <li>f. In the <b>Choose template document</b> section, click <b>Upload</b>.</li> <li>g. In the <b>Upload file</b> window, click <b>Choose file</b>, and then navigate to the text template.</li> <li>h. Click <b>Open</b>.</li> <li>i. Click <b>Done</b>.</li> </ul>
<b>Use an existing template</b>	<ul style="list-style-type: none"> <li>a. In the <b>Choose template document</b> list, select a template that you want to use.</li> </ul>

5. In the **Document name** field, enter a descriptive name for the output document.

6. In the **Category** list, select a category for your document.

7. Click **Save**.

At run time, when your application generates a case document, users can open and download it in the **Recent content** section.

Creating project documents for stakeholders ([on page 113](#))

Exporting a data model into a document ([on page 113](#))

## Case documents tags

Customize your case documents by configuring the tags that you include in your case document templates. For example, if your case includes financial information, you can include tags for the currency.

The following table lists tags that you can insert in case document templates, with configuration parameters and examples:

Type	When to customize	Configuration parameters	Example
Date and time		"name": ".Property-Name", "type": "P"	Design time: <pegaref con-



Type	When to customize	Configuration parameters	Example
	Display date only	"format":"MM/dd/yyyy"	<pre>fig={ "name": ".Date- TimeOfRe- port", "for- mat": "MM/dd/yyyy hh:mm:ss a", "type": "P", set- TZ="true" }&gt;</pre> <p>Run time: 09/19/2019 11:30:15 AM GMT</p>
	Display date and time, not 24h format	format:"MM/dd/yyyy HH:mm:ss a"	
	Display date and time, 24h format	format:"MM/dd/yyyy HH:mm:ss"	
	Display timezone	setTZ:"true"	
	Do not display time- zone	setTZ:"false"	
Date only		"name": ".Property- Name", "type": "P", "for- mat": "MM/dd/yyyy"	<pre>Design time: &lt;pe- garef con- fig={ "name": ".Da- teOfReport", "for- mat": "MM/dd/yyyy", "type": "P"</pre> <p>Run time: 09/19/2019</p>
Time only		"name": ".Property- Name", "type": "P"	<pre>Design time: &lt;pe- garef con- fig={ "name": ".Time- OfReport", "for- mat": "hh:mm a", "type": "P", "set- TZ": "false" }&gt;</pre> <p>Run time: 04:13 PM</p>
	Not 24h format	"format": "hh:mm a"	
	24h format	"format": "HH:mm"	
	Display timezone	setTZ:"true"	
	Do not display time- zone	setTZ:"false"	
Integer, decimal, dou- ble		"name": ".Property- Name", "type": "P"	<pre>Design time: &lt;pe- garef con- fig={ "name": ".To- talCost", "s- cale": "2", "type": "P", "do- Group- By": "true", "group-</pre>
	Display decimal places	"scale": " <i>Any number be- tween 0 and 10</i> "	
	Group digits	"doGroup- By": "true", "groupBy-	

Type	When to customize	Configuration parameters	Example
		Size:"Any number between 2 and 5"	BySize : "3", "isCurrency" : "true", "currency" : "\$_val (USD) " }>
	Use symbol for currency	"isCurrency":"true","currency":"Symbol_val"	Run time: \$12,345,678.00 (USD)
	Use code for currency	"isCurrency":"true","currency":"_val (Code)"	
	Use symbol and code for currency	"isCurrency":"true","currency":"Symbol_val (Code)"	
Boolean		"name":".Property-Name","type":"P","tValue":"ValueForTrue","fValue":"ValueForFalse"	Design time: <pegaref config={ "name": ".Property-Name", "type": "P", "tValue": "Yes", "fValue": "No" }>  Run time: Yes
Attachment		"name":".AttachmentFieldProp.pxResults(1).pyAttachmentLink","type":"A"	Design time: <pegaref config={ "name": ".AttachmentFieldProp.pxResults(1).pyAttachmentLink", "type": "A", "showLabel": "true", "attachLabel": ".XYZ", "width": "20", "height":  Run time: The application displays image attachments, and em-

Type	When to customize	Configuration parameters	Example
			beds files in a .docx format as inline attachments. Otherwise, the application adds attachments as objects by using object linking and embedding (OLE).
Text		"name": ".PropertyName", "type": "P"	<p><b>Note:</b> Users can insert images of any size when they use a rich text editor.</p> <p>Design time: <code>&lt;pegaref config=fig={ "name": ".PropertyName", "type": "P", "format": "TA" }&gt;</code></p> <p>Run time: Users can enter text in a text area.</p>
	Text area	"format": "TA"	
	Rich text editor	"format": "RTE"	
Properties inside page list		"name": ".PageList().PropertyName"	<p>Design time: <code>&lt;pegaref config=fig={ "mode": "page-list", "name": ".PageList().PropertyName", "type": "P" }&gt;</code></p> <p>Run time: Row 1: <i>Value1</i>, Row 2: <i>Value2</i>.</p>

Type	When to customize	Configuration parameters	Example
Properties inside page		"name": ".Page.PropertyName"	<p><b>Note:</b> Document generation supports embedded page levels, for example .Page1-.Page2-.Page3.PropertyName.</p> <p><b>i</b></p> <p>Design time: <code>&lt;pegaref config={ "name": ".Page-PropertyName", "type": "P" }&gt;</code></p> <p>Run time: <i>PropertyValue</i></p>
Section		"name": "Section-Name", "type": "S"	<p><b>Note:</b> Document generation does not support skin styling and CSS3. As a result, the application might not properly render Pega autogenerated sections.</p> <p><b>i</b></p>

Type	When to customize	Configuration parameters	Example
			<p>Design time: <code>&lt;pegaref config={ "name": "Section-Name", "type": "S" }&gt;</code></p> <p>Run time: <i>Section name</i></p>
Correspondence		"name":"CorrespondenceName","type":"C"	<div data-bbox="1149 653 1461 1268" style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p><b>Note:</b> Document generation does not support skin styling and CSS3. As a result, the application might not properly render Pega autogenerated sections.</p> </div> <p>Design time: <code>&lt;pegaref config={ "name": "CorrespondenceName", "type": "C" }&gt;</code></p> <p>Run time: <i>Correspondence content</i></p>

Generating case documents (on page 112)

## Case documents specifications

Ensure that your application generates the necessary documents in a case life cycle by learning more about the specifications of the elements that you can include in your case documents. You can then save time by generating documents that do not display error messages, and understand how you can achieve your business goals by including relevant business data in the output documents.

## Rich text editor

When you include text rich editors in your documents, consider the following factors:

- Documents only support rich text editors (RTEs) that you embed in a top-level paragraph alone. Documents do not support RTE that you insert in the following places:
  - Inside lists and tables
  - Inside paragraphs that contain text or expressions
- RTEs might have issues with processing large images. To avoid this problem, use images with a smaller width and height.
- RTEs that you render in a browser, and then embed inside a .docx document, might look different.
- If an output document includes an RTE, you cannot convert it to .pdf format with tools such as Apache POI, IText, or FOP.

## Links

Documents support only static links with dynamic text.

## Attachments

- When you use MS Office on Mac OS to create a template, the application processes only Object Linking and Embedding (OLE) that are images.
- The output case document is generated in .docx format, in which images and attachments are embedded as inline attachments. The document generates other attachments as icons with attachment names. Users can then locate the attachments in the **Recent content** section.
- Documents only support the *Data-WorkAttach-File* and *Rule-File-Binary* rule types.
- Applications can source attachments from a Pega database or any external storage that Pega Platform supports.

## Text area

- Documents treat text areas as standard text, however, this approach requires a lot of API processing to maintain line breaks. Too many text areas might slow down generation of the document.
- For faster processing, use a text area field by itself, within a paragraph.





## Nested paragraphs

Documents support only scalar properties inside nested paragraphs.

## Tables

- One row can only contain a single page list property. For example, one row can only have the `.pxResults().pyLabels` or `.pyResults().pyLabel` property. If you create repeating rows, reference the columns from the same page list.
- Documents support only image attachments in tables.
- Documents do not support HTML in tables.

## Headers and footers

- Documents support only image attachments in headers and footers.
- Documents do not support HTML in headers and footers.

## Images

- Large images might cause issues in output documents. To avoid these issues, use the width and height key in a JSON expression when you embed an image. Size values are measured in pixels.
- Images in `.gif` and `.svg` formats do not work consistently with document generation. To avoid issues, use images in other formats.

---

[Generating case documents \(on page 112\)](#)

## Case documents error messages

Resolve any issues when you generate case documents by understanding the error messages. Inline error messages that an output document displays might help you locate and fix the issue.

The following table lists the error messages that are associated with case document generation, with explanations:

Error message	Description
ERR_SKIP_HTML_IN_TABLE	A table cannot contain content from a section, correspondence, a rich text editor, or HTML.
ERR_SKIP_HTML_IN_HEADER/FOOTER	A header or footer cannot contain content from a section, correspondence, a rich text editor, or HTML.



Error message	Description
ERR_RETRIEVING_HTML	A document cannot retrieve HTML from a section or correspondence.
ERR_INVALID_CONFIGURATION	The JSON configuration is invalid.
ERR_INVALID_TYPE	The JSON type key is not a property, attachment, section, or correspondence.
ERR_EMPTY_PROPERTY_NAME	The JSON name key is empty.
ERR_SKIP_ATTACHMENT_IN_PARA	A document cannot contain an attachment in a paragraph with text.
ERR_INVALID_ATTACHMENT_INFO	The JSON configuration is correct but the attachment keys are invalid.
ERR_SKIP_NON_IMAGE_ATTACHMENT_IN_TABLE	A table cannot contain non-image attachments.
ERR_SKIP_NON_IMAGE_ATTACHMENT_IN_HEADER/FOOTER	A header or footer cannot contain non-image attachments.
ERR_COULD_NOT_ADD_ATTACHMENT	An application could not add an attachment to a document.
ERR_ATTACHMENT_NOT_FOUND	An application could not find an attachment that matches a provided key.
ERR_ADDING_IMAGE	An application failed to add an image to a document.
ERR_PROCESSING	An error occurred during processing of HTML or an attachment.
ERR_PAGELIST_UNSUPPORTED	A document does not support page lists outside tables.

[Generating case documents \(on page 112\)](#)

## Activities for integrating cases with Excel documents

To integrate case data with external resources, call the *pxParseExcelFile* and *pxGenerateExcelFile* activities that Pega Platform provides. By calling these activities, you can import and export case data by using Excel files. For example, if a customer in a bank loan dispute provides income information in an Excel file, you can use the *pxParseExcelFile* activity to read the file and integrate the data into the case processing.



The *pxParseExcelFile* and *pxGenerateExcelFile* activities are both final activities and APIs.

## The *pxParseExcelFile* activity

The following table lists the features that Pega Platform supports when parsing Excel files with the *pxParseExcelFile* activity:

Feature	Name	Support- ed in Pega Platform 8.1	Support- ed in Pe- ga Plat- form 8.2 and later	Description
Supported file format	.xlsx	Yes	Yes	N/A
Row-level parsing		Yes	Yes	The application parses an uploaded file row by row and replaces every cell value with a corresponding value from the clipboard.
Sheet-specific parsing		Yes	Yes	The application parses specific sheets within an Excel file after you parse the required sheet names as a comma-separated parameter. Otherwise, the application parses all of the sheets.
Supported Excel cell types	String	Yes	Yes	N/A
	Numeric	Yes	Yes	N/A
	Boolean	Yes	Yes	N/A
	Formula	No	Yes	N/A
	Date	No	Yes	N/A
Supported clipboard data types	String	Yes	Yes	N/A
	Integer	Yes	Yes	N/A
	Double	Yes	Yes	N/A
	Date	Yes	Yes	N/A
	Boolean	Yes	Yes	N/A
Properties from any clipboard page, such	Page	Yes	Yes	Pega Platform supports only single-level page lists, for example in the following format: <code>{.pxResults().value input}, {.description input}</code>



Feature	Name	Support- ed in Pega Platform 8.1	Support- ed in Pe- ga Plat- form 8.2 and later	Description
as a re- port defin- ition or da- ta page	PageList	Yes	Yes	
Styles	Font, cell, color	Yes	Yes	The application ignores any cell styles while parsing data to the clipboard.

The following table lists the parameters that Pega Platform supports when parsing Excel files with the *pxParseExcelFile* activity:

Parameter	Support- ed in Pega Platform 8.1	Support- ed in Pega Platform 8.2 and lat- er	Description
<i>FSFileName</i>	Yes	Yes	The name of the file to parse.
<i>TemplateRFB</i>	Yes	Yes	Template details in the <i>Fileformat!Filename!xlsx</i> format.
<i>sheetsTo- Parse</i>	Yes	Yes	Sheet names stored as comma-separated values. If the parameter is empty, the application skips validation and parses all of the sheets.
<i>sheetWiseDa- ta</i>	Yes	Yes	When checked, the application organizes parsed data sheetwise by using the <i>pySheets(sheet name)</i> page group.
<i>hasHeader</i>	Yes	Yes	The application returns a list of template sheets with a header that has two rows – a header and a property information row, and a list of templates that have headers but no property information row.
<i>bDeleteFile</i>	Yes	Yes	When the value is <code>true</code> , the application deletes the file after parsing.

Parameter	Support- ed in Pega Platform 8.1	Support- ed in Pega Platform 8.2 and lat- er	Description
<i>outputPage- Name</i>	Yes	Yes	When you define this parameter, the application saves all the error, warning, and information messages on this page. When you leave the parameter blank, the application saves the messages on the primary page.

## The *pxGenerateExcelFile* activity

The following table lists features that Pega Platform supports when generating Excel files with the *pxGenerateExcelFile* activity:

Feature	Name	Support- ed in Pe- ga Plat- form 8.1	Support- ed in Pe- ga Plat- form 8.2 and later	Description
Support- ed file format	.xlsx	Yes	Yes	N/A
Col- umn-lev- el gener- ation		Yes	Yes	The application generates a file column by column, supporting both homogenous and non-homogenous types of data, for example, columns with an unequal numbers of rows.
Support- ed tem- plate type	Rule File Binary (RFB)	Yes	Yes	Template sheets with a header need to have two rows – a header and a property information row. Template sheets without a header row still need to have a property information row.
Support- ed Excel cell types	String	Yes	Yes	
	Numeric	Yes	Yes	
	Boolean	Yes	Yes	
	Formula	No	Yes	
	Date	No	Yes	

Feature	Name	Support- ed in Pe- ga Plat- form 8.1	Support- ed in Pe- ga Plat- form 8.2 and later	Description
	Hyperlink	No	Yes	Sample cell value: <code>{.pxResults().linkToSheet HYPERLINK}</code> where <i>linkToSheet</i> is a text value in the following for- mat: <code>".myLabelPropery    .myURLProperty" hyperLinkType</code> is one of the values: EMAIL, URL, SHEET or <empty>.  For example: <code>".pyID   URL  .pyURLContent"</code>
Support- ed clip- board da- ta pages	Text	Yes	Yes	
	Decimal	Yes	Yes	
	Double	Yes	Yes	
	Integer	Yes	Yes	
	Date	Yes	Yes	
	TimeOf- Day	Yes	Yes	
	DateTime	Yes	Yes	
	TrueFalse	Yes	Yes	
Proper- ties from any clip- board page	Page	Yes	Yes	Pega Platform supports only single-level page lists, for example in the following format: <code>{.pxResuls().value in- put} , {.description input}</code>
	PageList	Yes	Yes	
Styles	Font, cell, color	Yes	Yes	The application clones styles from a template and ap- plies them to respective column cells in the generated files.

The following table lists parameters that Pega Platform supports when generating Excel files with the `pxGenerateExcelFile` activity:

Parameter	Support- ed in Pega Platform versions 8.1 to 8.3	Support- ed in Pega Platform 8.4 and lat- er	Description
<i>FSFileName</i>	Yes	Yes	The name of the file to download.
<i>TemplateRFB</i>	Yes	Yes	Template details in the <i>Fileformat!Filename!xlsx</i> format.
<i>DownloadFile</i>	Yes	Yes	If the value is <code>true</code> , the application downloads the generated file.
<i>DeleteFileAfterDownload</i>	Yes	Yes	If the value is <code>true</code> , the server deletes the generated file after the application downloads the file.
<i>evaluateFormulasLater</i>	No	Yes	If the value is <code>true</code> , advanced formulas defined in the template evaluate after you download and open the generated Excel file. Otherwise, the server evaluates the formulas in the template by using the Apache POI library (subject to support by the library).


### Related information

[Case documents tags \(on page 114\)](#)

[Integrating a case with an external resource \(on page 374\)](#)

## Tags for generating .docx documents through APIs

APIs support generating formatted and personalized documents that you can use during case processing. To save time and resources, the *pxGenerateWordDocument* API provides the possibility of generating a .docx document on the server side, without installing additional software, such as Silverlight.

**Note:** Word Merge is deprecated in align with deprecation of Silverlight. Instead, use the  *pxGenerateWordDocument* API to generate .docx documents. For more information about supported functionalities, see [Pega Platform Client Operating System and Browser Support](#).

To switch from the Silverlight software to the *pxGenerateWordDocument* API, move your Silverlight documents templates to the new server-side document generating format. The following table provides information about support for Silverlight tags in the *pxGenerateWordDocument* API:



Tag in Silverlight	Support in the <i>pxGenerateWord-Document API</i>	Comments
<code>{pega:reference propertyreference }</code>	Yes	<ul style="list-style-type: none"> <li>• Data references can vary as Silverlight reads data from BLOBs.</li> <li>• No support for localization in <i>pxGenerateWord-Document</i></li> </ul>
<code>{pega:attachment attachfile="property-Name.insertMethod.extension.pzInsKey" }</code>	No	<ul style="list-style-type: none"> <li>• Use the tag <code>&lt;pegaref config={"name":"AttachmentLink","type":"A"}&gt;</code> instead.</li> <li>• <i>AttachmentLink</i> is the property reference that provides the <i>pzInsKey</i> of Data-Work-Attach-File or Rule-File-Binary.</li> <li>• No support for attachments other than images inside tables, footers, and headers.</li> </ul>
<code>{pega:insert corr=corrname}</code>	Partial	<ul style="list-style-type: none"> <li>• No support for correspondence inside tables, headers and footers.</li> </ul>
<code>{pega:insert corrinline=corrname}</code>	No	<ul style="list-style-type: none"> <li>• Convert <code>{pega:insert corrinline=corrname}</code> to <code>{pega:insert corr=corrname}</code>.</li> </ul>



Tag in Silverlight	Support in the <i>pxGenerateWordDocument</i> API	Comments
		<ul style="list-style-type: none"> <li>• Alternatively, to add correspondence in-line within a document, use <code>&lt;pegaref config={"name":"Correspondence-Name","type":"C"}&gt;</code>.</li> <li>• No support for correspondence inside tables, headers and footers.</li> </ul>
<code>{pega:binary imagefile= AppName.image-FileName.FileType}</code>	No	<ul style="list-style-type: none"> <li>• Parse images by using <code>&lt;pegaref config={"name":"AttachmentLink","type":"A"}&gt;</code>.</li> <li>• <code>AttachmentLink</code> is the property reference the provides the <code>pzInsKey</code> of <code>Data-Work-Attach-File</code> or <code>Rule-File-Binary</code>.</li> </ul>
<code>{pega:binary binaryfile=Template-Class.TemplateName}</code>	No	No support for reading the template and parsing the tags inside the template, and then adding the template back to the target file.
<code>{pega:binary imagefilelocalpath="LocalPath" }</code>	No	N/A
<code>{pega:binary imagefileserverpath="ServerPath" }</code>	No	N/A

The specifications of the *pxGenerateWordDocument* API are as follows:



- The *pxGenerateWordDocument* API supports templates in .docx format only.
- No support for macros inside templates.
- The output format of the documents that you generate through the *pxGenerateWordDocument* API is always .docx.
- No support for client-side editing of documents that you generate through the *pxGenerateWordDocument* API.
- The *pxGenerateWordDocument* API creates documents in the following ways:
  - The *pxGenerateWordDocument* API adds .docx attachments inline.
  - The *pxGenerateWordDocument* API adds images inline and does not support the .svg images.
  - The *pxGenerateWordDocument* API adds attachments other than .docx format and images in the .svg format as OLE objects inside the generated .docx output document.
  - Inside tables, headers, and footers, the *pxGenerateWordDocument* API supports the addition of image attachments only.

## Moving a case to a different stage

Customize the order in which a case proceeds by using the Change to a stage shape to move a case to a different stage in the case life cycle, or the Change to the next stage shape to move a case to the next stage. By revisiting or skipping a stage, you can support out-of-sequence processing in a case. For example, when a customer changes jobs during a loan request case, the case returns to a stage in which the case worker reviews the loan risk.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. Add the Change to a stage shape or the Change to next stage shape to the life cycle of your case:

### Choices

**Automatically move the case a specific stage**

### Actions

- a. In the stage from which you want to move the case, click **Step > More > Automations > Change to a stage**.
- b. Click **Select**.
- c. In the **Step** properties pane, in the **Stage** list, select the stage that you want the case to enter.
- d. **Optional:** To provide additional comments about the change, in the **Audit note** field, enter some helpful extra text.
- e. Click **Save**.

Choices	Actions
<p><b>Automatically move the case to the next stage</b></p>	<ul style="list-style-type: none"> <li>a. In the stage from which you want to move the case, click <b>Step &gt; More &gt; Automations &gt; Change to next stage</b>.</li> <li>b. Click <b>Select</b>.</li> <li>c. <b>Optional:</b> To provide additional comments about the change, in the <b>Audit note</b> field, enter some helpful extra text.</li> <li>d. Click <b>Save</b>.</li> </ul>

3. Click **Save**.

At run time, the system resolves open assignments before the case leaves the current stage. All processes that originated from the stage, including supporting processes and excluding spin-off flows, stop.

[Stages in a case life cycle \(on page 46\)](#)

[Steps in a case life cycle \(on page 78\)](#)

## Pausing and resuming processes in cases

For greater flexibility and improved case resolution, pause and resume processes when a case meets conditions that you define. When you create dependencies between your cases, the ability to pause processes can facilitate the resolution of complex cases that require other processes to end.

For example, you can create a parent case that describes a hiring process of a job candidate, and then create child cases that represent running a background check and collecting all of the required documents. You can configure the parent case to pause until a case worker resolves all of the child cases or any of the child cases.

For relevant training materials, see the [Pausing and resuming case processing](#) module on Pega Academy.

1. Add a Wait shape to the life cycle of your case:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - c. In a stage in which you want to add a Wait shape, click **Step > More > Automations > Wait**, and then click **Select**.
2. Define the conditions for resuming the process:





**Note:** A **Wait** shape that you configure with advanced options can only be modified in Dev Studio.

### Choices

#### Wait until all child cases reach resolution

#### Wait until one or all child cases reach resolution

### Actions

- a. In the **Wait type** list, select **Case dependency**.
- b. Select the **Wait for all child cases to be resolved** check box.


- a. In the **Wait type** list, select **Case dependency**.
- b. In the **Wait for (Case type)** lists, define how many of the selected case types need to reach resolution before the process resumes.
- c. Click **To be resolved**.



**Note:** A case reaches resolution when its case status starts with the word "Resolved".

- d. **Optional:** To consider statuses only after the case reaches the Wait shape, select the **Consider status after wait begins** check box.
- e. If the case type has more than one parent in your case type hierarchy, in the **Scope** list, select an option that identifies all the instances on which you want the process to wait.

Choices	Actions
<p><b>Wait until one or all child cases reach a specific status</b></p>	<div data-bbox="941 220 1468 546" style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; background-color: #E1F5FE;"> <p><b>Tip:</b> To select a case type that is not a current case, a parent case, or a top-level case, in the <b>Scope</b> list, select <b>Other</b>, and then in the <b>Class</b> field enter the class that defines the case type.</p> </div> <ol style="list-style-type: none"> <li>a. In the <b>Wait type</b> list, select <b>Case dependency</b>.</li> <li>b. In the <b>Wait for (Case type)</b> lists, define how many of the selected case types need to reach a certain status before the process resumes.</li> <li>c. Click <b>To reach status</b>.</li> <li>d. In the <b>To reach status</b> field, press the Down arrow key, and then select a status value.</li> <li>e. <b>Optional:</b> To consider statuses only after the case reaches the <b>Wait</b> shape, select the <b>Consider status after wait begins</b> check box.</li> <li>f. If the case type has more than one parent in your case type hierarchy, in the <b>Scope</b> list, select an option that identifies all the instances on which you want the process to wait.</li> </ol> <div data-bbox="941 1680 1468 1837" style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-top: 2px solid #0070C0; padding: 10px; background-color: #E1F5FE;"> <p><b>Tip:</b> To select a case type that is not a current case, a parent case, or a top-level case, in the</p> </div>

Choices	Actions
	<p> <b>Scope</b> list, select <b>Other</b>, and then in the <b>Class</b> field enter the class that defines the case type.</p>
<b>Wait until a set amount of time passes</b>	<ol style="list-style-type: none"> <li>In the <b>Wait type</b> list, select <b>Timer</b>.</li> <li>Click <b>Set date/time interval</b>.</li> <li>Enter integers in one or more of the fields that appear, based on the interval of time that you want the process to wait.</li> </ol>
<b>Wait until a specific date or time</b>	<ol style="list-style-type: none"> <li>In the <b>Wait type</b> list, select <b>Timer</b>.</li> <li>Click <b>Reference date/time</b>.</li> <li>In the <b>Date/Time</b> field, select a property that stores the date and time when you want the process to resume, or click the <b>Build an expression</b> icon next to the field to build an expression that returns a date and time. Ensure that you select a property that references the date and time in the future.</li> </ol>

- Optional:** To allow case workers to cancel the wait conditions and continue processing a case, select **Users can choose to continue process**.
- Click **Save**.

[Case life cycle elements \(on page 43\)](#)

[Assignment shapes in processes \(on page 270\)](#)

[Building expressions with the Expression Builder \(on page \)](#)

## Persisting temporary cases

Reuse business processes by persisting temporary cases into permanent objects in your database. By using this action as a part of your duplicate tracking logic, you can save resources and ensure that your application creates only unique, relevant cases.



You can create a temporary case to save your database resources when you are unsure whether the case is unique to your application or when a customer discards the case before completing it. After you confirm that the case is unique or decide to reuse the case in the future, you can persist a temporary case. You can also use temporary cases before the system evaluates whether the interaction with a customer requires creating a case. For example, in an airline reservation application, when a customer connects with the system, the application creates a temporary case. If the customer simply reviews their scheduled flights, the application closes the temporary case after the customer leaves the system. If the customer wants to edit their reservation, the application persists the case.

1. Add the Persist Case shape to your case life cycle:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - c. In a process in which you want to add the shape, click **Step > More > Automations > Persist case**.
  - d. Click **Select**.
2. In the **Persist case** field, enter a name for the step.
3. **Optional:** to provide more auditing information, in the **Step** properties panel, enter an audit note.
4. To provide a link to this step from bread crumb trail navigation, select **Enable navigation link**.
5. Click **Save**.

---

[Searching duplicate cases \(on page 109\)](#)

[Flow shapes \(on page 251\)](#)

[Adding an automated step to a process \(on page 101\)](#)

## Posting messages to a case

Inform case workers about updates to a case and provide a visible discussion thread for stakeholders by posting Pulse messages to a case. By posting messages, you provide additional relevant information that enriches case processing and helps you move your cases faster towards resolution. For example, when reviewing a job candidate, you can post a message that informs other involved stakeholders about the result of the recruitment process.

1. Add the Post to Pulse shape to the life cycle of your case:
  - a. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the process where you want to post Pulse messages, click **Step > More > Automations > Post to Pulse**.



- c. Click **Select**.
  - d. **Optional:** To provide a unique name for the step, in the **Post to Pulse** field, enter a value.
2. In the **Step** properties pane, identify the user who is responsible for the post:
    - To create a post for the user who currently processes the case, in the **User posting** list, select **Current operator**.
    - To create a post for another user, in the **User posting** list, select **Other**, and then select the ID of the user to post the message.
  3. In the **Message** field, enter the text for the post.
  4. **Optional:** To provide more information for auditing purposes, in the **Audit note** field, enter additional text.
  5. Click **Save**.

[Case life cycle elements \(on page 43\)](#)

[Collaborating with users by using Pulse \(on page 499\)](#)

## Preloading a data page

Complete work faster by preloading data that your application displays to users during case processing. By preloading a data page, you reduce the time that an application needs to display information to users. For example, a financial application can preload a data page that stores currency rates so that users can quickly obtain the information that their business case requires.

In your case life cycle, add the Load data page step before the step that displays information to users. For example, add the Load data page step that preloads currency rates before the Calculate interest rate step in which a customer service representative (CSR) calculates rates for customers who want to obtain a loan in another currency.

1. Add the Load data page step to a case life cycle:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In a process where you want to preload a data page, click **Step > More > Automations > Load data page**.
  - c. Click **Select**.
  - d. To provide a unique name for the step, in the **Load Data page** field, enter a value.





2. In the **Step** properties pane, in the **Data Page Name** field, press the Down arrow key, and then select the data page that you want to preload.
3. If the data page contains parameters, in the **Parameters** section, specify values for the parameters.
4. **Optional:** To provide more information for auditing purposes, in the **Audit note** field, press the Down arrow key, and then select an audit note to associate with the step.
5. **Optional:** To enable users to navigate to this step from the case breadcrumb navigation, select the **Enable navigation link** check box.
6. Click **Save**.

---

Data page definition (on page )

Creating a data page (on page )

Configuring load management options for data pages (on page )

## Running a queued background process in a case

Speed up case resolution and automate actions in your cases by running background processes. You can use the Run in background shape to start a background process, such as asynchronous processing or queuing items.

For example, by implementing queue processors, your application can send notifications to users and calculate information such as the list of recipients, the message, and the channel asynchronously to the flow of the case. As a result, case processing can continue before the background process ends.

The Run in background shape uses a Queue Processor rule for background processing. You can select a standard queue processor rule that Pega Platform includes by default, or you can use a custom, dedicated queue processor rule.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Workflow** tab, click **Life cycle**.
3. Hover over a process in a stage, and then click **Step > More > Automations > Run in background**.
4. Click **Select**.
5. In the **Step** properties pane, in the **Type of queue** list, choose a type of the queue processor that you want to run:
  - To use a default queue processor, select **Standard**, and then in the **Activity name** field, provide an activity that the rule runs.
  - To use a custom queue processor rule, select **Dedicated**, and then in the **Queue processor** field, provide the queue processor that you want to run.

For more information about when to use standard or dedicated queues, see Queue processor rules (on page ):

6. In the **Lock using** list, select a type of lock that you want to apply:



- To open a top-level page that your application currently processes, select **Primary page**.

Your application uses a *pzlnskey* to find and lock the page.

- To open a page that you define manually, select **Key defined on property**, and then in the **Property name** field, enter a property name that corresponds with the page that you want to open and lock.
- To open a page without a lock from the database, select **None**.

7. **Optional:** To resolve the activity in a different context than the context that your application uses to queue the item, in the **Alternate access group** field, enter the name of an access group that you want to use to resolve the activity.

8. Click **Save**.

---

Queue processor rules (*on page* )

Replacing an agent with a queue processor rule for delayed messages (*on page* )

Replacing an agent with a queue processor rule for real-time messages (*on page* )

Changing a queue processor rule state (*on page* )

Tracing a queue processor (*on page* )

## Running a questionnaire in a case

Collect a wide variety of information from your application users by running a questionnaire in a case. By providing questions and collecting answers in a structured format, you can quickly incorporate user feedback into a case. Reusing questionnaires speeds up the development of your application and helps you deliver accurate applications that precisely meet the business needs of the users.

Your application runs the questionnaire as a child case in the context of the parent case that includes the Questionnaire step. To save more time, you can propagate data from the parent case to the questionnaire. For example, when a user provides basic information about their stay at a hotel by completing a booking form in a parent case, you can propagate the obtained data to an exit questionnaire. As a result, the questionnaire already includes their basic information, such as the dates of their stay and type of accommodation.

Data propagation also helps you deliver more accurate applications, as you can use data from a parent questionnaire to build logic in a child questionnaire. For example, if a user provides a monthly income greater than a specified amount in a parent questionnaire, you can create visibility conditions in the child questionnaire to skip further questions about their income.



1. Add the Questionnaire shape to the life cycle of your case.
  - a. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - b. On the **Workflow** tab, in the **Case life cycle** section, hover over a process in a stage, and then click **Step > More > Automations > Questionnaire**.
  - c. Click **Select**.
2. In the **Create Questionnaire** field, provide a name for the step, for example, `Rate customer service`.
3. In the **Step** properties pane, in the **Select questionnaire** list, select the name of the questionnaire that you want to run as a child case.
4. **Optional:** To reuse information from the parent case, transfer data from the current case to the questionnaire:

Choices	Actions
<p><b>Map fields from the case type to the questionnaire</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>Step</b> properties pane, select the <b>Transfer info to questionnaire case</b> check box.</li> <li>b. In the <b>Transfer information</b> dialog box, in the <b>'From' field</b> column, select the field with the values that you want to transfer.</li> <li>c. In the <b>'To' field</b> column, press the Down arrow key, and then select a destination field to which you want to transfer the value.</li> <li>d. <b>Optional:</b> To add mapped fields to the case type view, in the <b>View</b> section, select the <b>Add mapped fields to [case type view name] view</b> check box.</li> <li>e. Click <b>OK</b>.</li> </ol>
<p><b>Configure advanced mapping by providing a data transform</b></p>	<ol style="list-style-type: none"> <li>a. Switch to Dev Studio.</li> <li>b. In the <b>Step</b> properties pane, select the <b>Transfer info to questionnaire case</b> check box.</li> <li>c. In the list of transfer options, select <b>Existing</b>.</li> </ol>

Choices	Actions
	<p>d. In the <b>Data transform</b> field, enter the name of the data transform that you want to apply.</p> <p>e. Click <b>Save</b>, and then switch to App Studio.</p>

At run time, when a user opens the questionnaire that is part of the case processing, the questionnaire already includes the specified data from the parent case.

- In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the questions in the questionnaire:
  - To route the questionnaire to the user who currently processes the case, select **Current user**.
  - To route the questionnaire to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to complete the questionnaire.
  - To route the questionnaire to a work queue that multiple users share, select **Work queue**, and then, in the **Work queue** field, enter the name of the work queue.
- Click **Save**.

After you add a Questionnaire step to your parent case type, you can access the data that users provide only after the child case for completing the questionnaire is complete. In the data model of the parent case, the system adds two new properties of a query type: a single property, and a list property. These two new properties source data from the questionnaire child case type. For more information about queries, see [Referencing a data page \(on page 161\)](#).

[Designing questionnaires \(on page 405\)](#)

[Sharing data between parent and child cases \(on page 217\)](#)

[Fine-tuning your questionnaire \(on page 426\)](#)

## Sending event notifications from cases

Notify users of your application about important case events by sending case notifications. You can send notifications when an event occurs in a case, for example, when the case reaches a particular stage. By defining different notifications in your application, you can communicate with stakeholders in a format that is specific to their channel, such as email, a web gadget, or a push notification. You can define push notifications and the subjects of email notifications in a dynamic way, so that the message is immediately meaningful to the users. For example, you can include a property that references a case ID. If you send email notifications, you can compose a message to meet your business needs, or you can reuse existing message templates to save time.



1. Add the **Send notification** shape to the life cycle of your case:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - c. In the stage in which you want to add a **Send notification** shape, click **Step > More > Automations > Send notification**, and then click **Select**.
2. In the **Step** properties panel, define the content and recipients of the notification:

Choices	Actions
<p><b>Create a new notification</b></p>	<ol style="list-style-type: none"> <li>a. Select <b>Create new</b>.</li> <li>b. In the <b>Recipients</b> list, select a type of recipient, and then define the recipient.</li> <li>c. <b>Optional:</b> To add more recipients, click <b>Add recipient</b>, and then repeat step <a href="#">2.b (on page 141)</a>.</li> <li>d. In the <b>Message</b> section, in the <b>Message/Email subject</b> field, press the Down arrow key, and then select the value that stores the push notification text or email subject, or enter text. You can reference property names to make the message more dynamic and meaningful.</li> <li>e. If you send email notifications, in the <b>Email content</b> section, click <b>Compose</b>, and then compose the email message. For more information about composing email messages, see <a href="#">Sending email notifications from cases (on page 236)</a>.</li> <li>f. If you send email notifications, in the <b>Email body</b> list, define the content of an email by selecting <b>Custom</b> and composing your message, or selecting <b>Cor-</b></li> </ol>

Choices	Actions
	<p><b>responsiveness</b> and choosing a correspondence template.</p> <p>For more information about composing email messages, see <a href="#">Sending email notifications from cases (on page 236)</a>.</p>
<b>Reuse existing notification</b>	<p>a. Select <b>Use existing</b>.</p> <p>b. In the <b>Notification name</b> list, select the notification that you want to send.</p>

3. Click **Save**.

[Steps in a case life cycle \(on page 78\)](#)

[Setting notification preferences \(on page 346\)](#)

[Notifications \(on page \)](#)

## Sending automatic emails from cases

By using email, you can share information about a case with stakeholders and case participants. Add the Send Email shape to your business process, to control the number of recipients and the message format. For example, you can inform stakeholders about the approval of a job candidate in a hiring process.

### Note:



When your application sends an email, the system updates the *pxSendDateTime* property on the *pxCorrSummary* work page that belongs to *pyWorkPage*. Such updates might cause issues with refreshing a case at run time. To prevent errors, the *pyEnableCorrSummary* when rule is set to false by default, and the system does not update the *pxSendDateTime* property. If your business scenario requires updating the *pxSendDateTime* property, in Dev Studio set the *pyEnableCorrSummary* when rule to true.

1. Add the **Send Email** shape to the life cycle of your case:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.

c. In the **Case life cycle** section, in a stage, click **Step > More > Automations > Send email**.

d. Click **Select**.

2. In the **Step** properties pane, identify the email recipients:

- To send the email to one recipient, in the **Send to** list, select **Email address**, and then enter the email address of a user who wants to receive information about your case.
- To send an email to more than one recipient, enter the email addresses and separate them with commas.
- To send the email based on the value of a field, in the **Send to** list, select **Field**, and then select the name of a field that stores an email address.
- To send the email based on the value of a user reference, in the **Send to** list, select **User reference**, and then select the name of a user reference from your data model.

At run time, this field stores information about one user only.

- To send the email to a group of stakeholders, in the **Send to** list, select **Participant**, and then select the name of a work party that defines a person, or a part of organization that is a participant of your case.

3. **Optional:** To add more recipients, click **Add recipient**, and then repeat step 2 (on page 143).

4. **Optional:** To send a shared email message to all recipients, select the **Send a common email to all recipients** check box.

If you leave the **Send a common email to all recipients** unchecked, an application sends separate email messages to each recipient and the recipients cannot view email addresses of other recipients.

5. In the **Subject** field, enter the title of the email by entering a string expression.

You can reference field property names in the title to make it more dynamic and meaningful, for example, "The case "+.pyID + " has been assigned to you".

6. Define your message content:

Choices	Actions
<b>Compose a new message</b>	a. In the <b>Message</b> list, select <b>Custom</b> .
<b>Use existing email content for the message</b>	a. in the <b>Message</b> list, select <b>Correspondence</b> . b. In the <b>Correspondence template</b> field, press the Down arrow key and select a template.

7. In the **Message content** section, define the email content by clicking **Compose**, and then editing the message in the rich text editor:

Choices	Actions
<b>Compose your own message</b>	<ol style="list-style-type: none"> <li>Enter and style the text by using the rich text editor toolbar.</li> </ol>
<b>Populate your email message with text from a template</b>	<ol style="list-style-type: none"> <li>In the <b>Compose message</b> window, click <b>Choose template</b>.</li> <li>Click a template that you want to use.</li> <li>Click <b>Select</b>. After the template text appears in your email message, you can edit this content.</li> </ol>
<b>Include the value of a field from your case in your email</b>	<ol style="list-style-type: none"> <li>In the rich text editor, position the text cursor in the place where you want to add a property.</li> <li>On the toolbar, click the <b>Insert property</b> icon.</li> <li>Select a property that you want to use.</li> </ol> <div data-bbox="938 1115 1466 1346" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 10px; margin-top: 10px;"> <p><b>Tip:</b> To prevent word clusters in your message, include a space before and after your property reference.</p> </div>
<b>Provide instant access to the case from your email</b>	<ol style="list-style-type: none"> <li>In the rich text editor, position the text cursor in the place where you want to add a link.</li> <li>On the toolbar, click the <b>Link</b> icon.</li> <li>Select the <b>Link to current case</b> check box.</li> <li>Enter the display text and title for the link.</li> <li>Click <b>OK</b>.</li> </ol>

8. If you create a custom message, in the **Message content** section, click **Compose**, and then edit the message in the rich text editor:



Choices	Actions
<b>Compose your own message</b>	<ol style="list-style-type: none"> <li>Enter and style the text by using the rich text editor toolbar.</li> </ol>
<b>Populate your email message with text from a template</b>	<ol style="list-style-type: none"> <li>In the <b>Compose message</b> window, click <b>Choose template</b>.</li> <li>Click a template that you want to use.</li> <li>Click <b>Select</b>. After the template text appears in your email message, you can edit this content.</li> </ol>
<b>Include the value of a field from your case in your email</b>	<ol style="list-style-type: none"> <li>In the rich text editor, position the text cursor in the place where you want to add a property.</li> <li>On the toolbar, click the <b>Insert property</b> icon.</li> <li>Select a property that you want to use.</li> </ol> <div data-bbox="938 1115 1463 1346" style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p><b>Tip:</b> To prevent word clusters in your message, include a space before and after your property reference.</p> </div>
<b>Provide instant access to the case from your email</b>	<ol style="list-style-type: none"> <li>In the rich text editor, position the text cursor in the place where you want to add a link.</li> <li>On the toolbar, click the <b>Link</b> icon.</li> <li>Select the <b>Link to current case</b> check box.</li> <li>Enter the display text and title for the link.</li> <li>Click <b>OK</b>.</li> </ol>

9. Click **Done**.

10. **Optional:** To add attachments to the email, select the **Include attachments** check box, and then select the files that you want to upload:

Actions	Choices
<b>Include all case attachments</b>	a. Select <b>All case attachments</b> .
<b>Include only specific case attachments</b>	a. Select <b>Choose attachments</b> .  b. select a category or a field that stores the attachments. Categories store groups of attachments with the same business classification, while fields store single attachments.  c. <b>Optional:</b> To add multiple attachments, click <b>Add attachment</b> , and then select another category or field.

11. Click **Save**.

At run time, when a case reaches the Send Email step, an application sends an email that you create to the recipients that you select.

[Adding an automated step to a process \(on page 101\)](#)

[Categorizing case attachments \(on page 457\)](#)

[Setting notification preferences \(on page 346\)](#)

## Sending push notifications from cases

To ensure timely resolution of cases, notify users of your application when a case requires action. By sending push notifications, you can inform users about events in a case at any time and place. Using an event-driven model to share information ensures that cases quickly reach their resolution.


1. Add the Push Notification shape to the life cycle of your case:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - c. In a stage in which you want to add a Push notification shape, click **Step > More > Automations > Push notification**, and then click **Select**.
  - d. Click **Select**.



2. In the **Step** properties panel, identify the recipients of the notification:

Choices	Actions
<p><b>Notify users of an application</b></p>	<p>a. In the <b>Recipient type</b> list, select <b>Application</b>.</p> <p>The system autopopulates the <b>To</b> section with the name of your application.</p>
<p><b>Notify users in an access group</b></p>	<p>a. In the <b>Recipient type</b> list, select <b>Role</b>.            b. In the <b>To</b> list, select an access group.            c. In the <b>To</b> field, press the Down arrow key, and then select an access group.</p>
<p><b>Notify a specific user</b></p>	<p>a. In the <b>Recipient type</b> list, select <b>Individual</b>.            b. In the <b>Recipient type</b> list, select <b>User</b>.            c. In the <b>To</b> list, select the identifier of the user that you want to notify.            d. In the <b>To</b> field, press the Down arrow key, and then select the identifier of the user that you want to notify.</p>

3. In the **Message** field, press the Down arrow key, and then select the property that contains the notification message that you want to send.

 **Tip:** To create meaningful notifications, use a combination of text and property references.

4. In the **Badge count** field, press the Down arrow key, and then select the property that stores the number of notifications.
5. **Optional:** To give users an audible notification when they receive a push notification, select the **Sound alert** check box.
6. **Optional:** To enable users on mobile devices to open the case directly from the notification, select the **Open a case in mobile app** check box.
7. Enter an audit note.
8. **Optional:** To provide a link to the Push notification step from a breadcrumb navigation link at run time, select the **Enable navigation link** check box.
9. Click **Save**.

---

[Case life cycle elements \(on page 43\)](#)

[Steps in a case life cycle \(on page 78\)](#)

[Engaging mobile users with push notifications \(on page \)](#)

## Saving data in a data page as part of a case life cycle

Manage data in your application more efficiently by saving case data to a data page. As a result, you can save the data in a separate object, and then manage the data independently of the case. Consequently, you speed up your application development and promote reuse of resources.

For example, in a financial application, if you save customer banking history as a data page, you can reuse the data in other rules or pages in your application that require the use of a data page.

The source of information that you want to reuse needs to be a savable data page or an autopopulated property that points to a savable data page. After updating information in a case, an application saves the information back to a selected data page.

When you save information to a data page, you can either call the data page directly or use an autopopulated property. Autopopulated properties prevent your application from monitoring changes in key parameters as they automatically retrieve the object as soon as any property value of the object is needed and a key to the object is available. Completing these actions improves the performance of your application.

1. Add the Save data page step to your case life cycle:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In a process where you want to preload a data page, click **Step > More > Automations > Save data page**.
  - c. Click **Select**.
  - d. **Optional:** To provide a unique name for the step, in the **Save data page** field, enter a value.
2. Configure how your application saves information to the data page:

Choices	Actions
<b>Save data directly to the data page</b>	<ol style="list-style-type: none"> <li>a. In the <b>Step</b> properties pane, in the <b>Data Page Name</b> field, enter the name of the savable data page.</li> <li>b. <b>Optional:</b> If the data page has parameters, in the <b>Parameters</b> section, specify values for the parameters.</li> </ol>
<b>Use an autopopulated property</b>	<ol style="list-style-type: none"> <li>a. In the <b>Step</b> properties pane, select the <b>Use associated property</b> check box.</li> <li>b. In the <b>Autopopulate Property Name</b> field, enter the name of the property that you want to use.</li> </ol>

3. **Optional:** To provide more information for auditing purposes, in the **Audit note** field, press the Down arrow key, and then select an audit note to associate with the step.
4. **Optional:** To enable users to navigate to this step from the case bread crumb navigation, select the **Enable navigation link** check box.
5. Click **Save**.

### Related information

Savable data pages ([on page](#))

Data page definition ([on page](#))

Data pages and parameters ([on page](#))

## Updating case information

You can use the Update Case shape to modify information, such as a description or priority, for one or more cases. By maintaining case information automatically, you can save time and resolve cases more quickly.

1. Add the Update Case shape to the life cycle of your case.
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - c. On the **Workflow** tab, click **Life cycle**.



d. In a stage, click > **More > Automations > Update a case.**



**Tip:** For advanced processes, click **Configure process** to add the shape to the flow canvas.

e. Click **Select.**

2. Indicate how many cases to update.

- To update a single case, click **A single case**, and then in the **With ID** field, enter a property reference or the unique identifier for an open case.
- To update the descendants of the current case, click **All child cases (and descendants).**
- To update a specific child case, click **Specific child case**, and then select a case to update.

3. In the **Data transform** field, select a data transform that sets property values in the case.

4. Indicate how many cases to update.

- To update all child cases, in the list, select **All.**
- To update a specific child case, select the case from the list.

5. Click the **Gear** icon to select target fields and values to apply from the current case.

6. Click **OK.**

7. Enter the audit note.

8. Click **Save.**

At run time, the data transform is applied to one or more cases. If an update to any child case fails, all changes made by the data transform are rolled back.

---

[Flow shapes \(on page 251\)](#)

[Adding an automated step to a process \(on page 101\)](#)

### Configuring a data model for a case

Configure a data model to define the data that you want to use in a case. For example, to include a user email address in a case, add an email address field to the data model.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Data model** tab, click **Add field.**



**Note:** You can also add new scalar fields to the data model from a view in the **User Interface** tab.

3. In the field configuration dialog box, in the **Field name** field, enter a descriptive name.
4. In the **Type** list, select a field type.
5. If more configuration options are available for the field type that you select, perform the following actions for a field:

Choices	Actions
<p><b>Add a text paragraph to the form</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>Type</b> list, select <b>Text (paragraph)</b>.</li> <li>b. Click the <b>Configure paragraph</b> icon.</li> <li>c. In the <b>Display as</b> list, define whether to display the paragraph as plain or rich text. <ul style="list-style-type: none"> <li>The following figure shows a rich text paragraph at run time that users can use to describe symptoms while booking a doctor's appointment in an application:</li> </ul> <div data-bbox="1071 1207 1502 1333" data-label="Image"> </div> </li> </ol>
<p><b>Add a picklist to the form</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>Type</b> list, select picklist.</li> <li>b. In the <b>Display as</b> list, define a display mode for the picklist.</li> <li>c. In the <b>Picklist options</b> list, define choices for the picklist. <ul style="list-style-type: none"> <li>You can create your own choices or source them from a data page.</li> </ul> </li> </ol>

## Choices

## Actions

- d. If you add your own choices, click **Add choice**, and then provide an option for users to select.
- e. If you use a data page to provide choices, select the data page that you want to use.

The following figure shows a picklist with radio buttons at run time that users can use to select an office location:

A screenshot of a picklist titled "Office location". It contains three radio button options: "Cambridge", "New York", and "Boston".

*Picklist*

## Add an attachment field to the form

- a. In the **Type** list, select **Attachment**.
- b. In the **Attachment category** list, select the category.

The following figure shows an attachment field at run time that users can use to add documents to a case:

A screenshot of an attachment field. The field is labeled "Documents" and contains the placeholder text "Drag and drop or choose file".

*Attachment field*

## Add a user reference to the form

- a. In the **Type** list, select **User reference**.
- b. In the **Display as** list, define whether users can search for a user ID by using a search box or a drop-down list.

The following figure shows a user reference field at run time configured as a search



Choices	Actions
<p><b>Add a field to capture data</b></p>	<p>box that users can use to select a doctor while booking an appointment in an application:</p> <div data-bbox="1084 365 1503 432" style="border: 1px solid gray; padding: 2px;"> <p>Your doctor Select...</p> </div> <p style="text-align: center;"><i>User reference field</i></p> <p>For more information about adding a field to capture data, see <a href="#">Creating fields for capturing data (on page 156)</a>.</p>

6. Click **Save**.

[Configuring primary fields \(on page 158\)](#)

UI components in a view (on page )

Viewing the data model for a case (on page )

Creating a Microjourney for customer success (on page )

## Referencing a case type

Build transparent and automatic relationships between case types by creating case references. When you reference a case type, you can pass data about one case type to another case type, and you can clearly view connections between case types and their instances at run time. You also save time and take advantage of low-code tools because you can view relations between case types without creating complex report definitions. For example, you can reference a `Manage account` case type from an `Offer opportunity` case type. As a result, you can quickly view offered opportunities that are related to a managed account. You can also create a UI to display the offered opportunities at run time from the view of a case for managing an account.

When you reference a case type, the system automatically creates a field of the query type in the data model of the referenced case type. For example, if you reference the `Manage account` case type in the `Offer opportunity` case type, the system automatically adds a query field in the data model of the `Manage account` case type. As a result, you can clearly analyze connections between case types.

To adjust the processing and display of case references at run time, you can decide whether a case references other cases as a single object or a list of objects. For example, you can reference only one run-time instance of a `Manage account` case type from a `Offer opportunity` case type. Then, an application displays only one case ID. You can also connect multiple instances of the `Offer opportunity` case type to the same instance of the `Manage account` case type, and configure an application to display a table with multiple case IDs.

**Note:** Case references work most efficiently in Cosmos React applications.

You can create case type references in the following elements of your application:

- Your case type data model. For more information, see [Configuring a data model for a case \(on page 150\)](#).
- A form in your case type for applications that do not use Cosmos React. For more information, see [Adding single-value fields to forms \(on page 174\)](#).
- Your application visual data model. For more information, see [Viewing an application data model \(on page 174\)](#).
- A data model of a data object. For more information, see [Viewing the data model for a data object \(on page 174\)](#).

1. Navigate to a place where you want to add a field:

Choices	Actions
<b>Add a field to a case type data model</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to a form in a view of a non-Cosmos React application</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>On the <b>Workflow</b> tab, in the <b>Case life cycle</b> section, click an assignment or an approval step.</li> <li>In the <b>Step</b> properties pane, click <b>Configure view</b>.</li> <li>In the dialog box, on the <b>Fields</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to an application visual data model</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Data</b>.</li> <li>In the <b>Data model</b> section, click <b>View</b>.</li> <li>In the data model, click a data object for which you want to create a data field.</li> <li>In the list of fields, click <b>Create new field</b>.</li> </ol>
<b>Add a field to a data object</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Data</b>.</li> <li>In the <b>Data objects</b> column, click the data object that you want to open.</li> <li>On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>

2. In the field configuration dialog box, for the **Field name** parameter, enter a unique label for the field.

3. In the **Type** list, select **Case reference**.



4. In the **Case type** list, select a case type that you want to reference.
5. In the **Options** list, define how you want to retrieve and display data from the source case type:
  - To create one entry for data from the source, select **Single record**.

For example, you can connect an *Opportunity* case type to one instance of the *Manage account* case type. At run time, an application displays only one field for the referenced data.

- To create a list of separate entries for data from the source, select **List of records**.

For example, you can connect a *Manage account* case type to multiple instances of an *Opportunity* case type. At run time, an application displays a grid table with sourced information.

6. **Optional:** To copy the referenced data into a case at run time, expand the **Advanced** section, and then select the **Reference data is copied to the case** check box.  
Copying data creates a snapshot of the referenced data and preserves the data in the target unchanged if data in the source changes.
7. **Optional:** To provide another data page to source data, expand the **Advanced** section, and then, in the **Data page** field, enter the data page that you want to use.
8. **Optional:** To change the unique identifier of a referenced case type, expand the **Parameters** section, and then configure how an application sources the new value:
  - To manually provide a new value, in the **pyID** list, select **Constant**, and then, in the text field, enter a new unique identifier for the case type.
  - To source the value from a field in the current case type, in the **pyID** list, select **Field**, and then, in the list, select a field that stores the value that you want to use as the unique identifier.
  - To source the value from a field in the referenced case type, in the **pyID** list, select **Field in target**, and then, in the list, select a field that stores the value that you want to use as the unique identifier.

You can reference a unique identifier, which is also a *pyID* parameter, in various elements in your application, such as reports.

9. Save your data field:
  - To save the field and instantly define another field, click **Submit & add another**.
  - To save your field and return to the configuration view, click **Submit**.

The system creates a field of a query type in the data model of the referenced case type, as shown in the following figure:

The figure shows two screenshots. The top screenshot is a dialog box titled "Add field to Offer opportunity". It contains the following fields and options:

- Field name \*: Manage account
- Type: Case reference
- Case type \*: Manage account
- Options:
  - Single record
  - List of records
- Advanced: >
- Buttons: Cancel, Submit & add another, Submit

The bottom screenshot shows the "Data model" configuration page for the "Manage account" case type. It features a search bar, a "Show system fields" checkbox, and a table of fields. The "OfferOpportunityListByManageAccount" field is highlighted with an orange border.

Name	ID	Type	Options
Case ID	pyID	Text (single line)	Key; Read-only
Description	pyDescription	Text (paragraph)	Plain text
Label	pyLabel	Text (single line)	
OfferOpportunityListByManageAccount	OfferOpportunityListByManageAccount	Query	Offer opportunity
Work status	pyStatusWork	Text (single line)	

Case reference field configuration

[Views for cases \(on page 168\)](#)

## Creating fields for capturing data

To decrease application development time and costs, reuse data objects across your application by creating fields that capture data. You can reference data between fields in several ways that help you to efficiently reuse resources without detailed knowledge about how the system stores data.

For example, in an `Account` data object that stores information that is related to customer accounts, you can create a `Contact` field to store contact information of a customer. To promote reuse, you can then reference a case type that the `Contact` field uses as the source of customer contact information.



**Note:** For applications that you build with the traditional Theme UI-Kit, see [Adding data references to forms](#).

You can create fields that capture data for the following interfaces:



- Your case type data model. For more information, see [Configuring a data model for a case \(on page 150\)](#).
- A form in your case type for applications that do not use Cosmos React. For more information, see [Adding single-value fields to forms \(on page 174\)](#).
- Your application visual data model. For more information, see [Viewing an application data model \(on page 174\)](#).
- A data model of a data object. For more information, see [Viewing the data model for a data object \(on page 174\)](#).

1. Navigate to a place where you want to add a field:

Choices	Actions
<b>Add a field to a case type data model</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to a form in a view of a non-Cosmos React application</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>On the <b>Workflow</b> tab, in the <b>Case life cycle</b> section, click an assignment or an approval step.</li> <li>In the <b>Step</b> properties pane, click <b>Configure view</b>.</li> <li>In the dialog box, on the <b>Fields</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to an application visual data model</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Data</b>.</li> <li>In the <b>Data model</b> section, click <b>View</b>.</li> <li>In the data model, click a data object for which you want to create a data field.</li> <li>In the list of fields, click <b>Create new field</b>.</li> </ol>
<b>Add a field to a data object</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Data</b>.</li> <li>In the <b>Data objects</b> column, click the data object that you want to open.</li> <li>On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>

- In the field configuration dialog box, for the **Field name** parameter, enter a unique label for the field.
- In the **Type** list, select **Data reference**.
- In the **Data object** field, select the data object that you want to use as the source.
- In the **Options** list, define how you want to reuse fields from the source:
  - To create one entry for all the fields from the source, select **Single record**.
  - To create a list of separate entries for the fields from the source, select **List of records**.

6. **Optional:** To provide additional information about your data field, expand and complete the **Advanced** section:
  - To provide a different ID than the autogenerated ID, in the **ID** field, enter a new value.
  - To provide additional information about the field, in the **Description** text box, enter additional details.
7. Save your data field:
  - To save the field and instantly define another field, click **Submit & add another**.
  - To save your field and return to the configuration view, click **Submit**.
8. For fields that you add to a form, in the **Options** list, define the display mode for the field:
  - To indicate that users can optionally complete the fields, select **Optional**.
  - To indicate that users need to provide a value in the fields, select **Required**.
  - To indicate that users can only view the fields, select **Read-only**.
9. Click **Submit**.

---

[Creating views for case types \(on page 173\)](#)

[Reusing fields on forms \(on page 179\)](#)

## Configuring primary fields

Define primary fields to automatically add the most relevant fields to your cases and reduce the amount of UI configuration that is necessary.



**Note:** Only applications that you build with Cosmos React support primary fields. For more information, see [Choosing a UI version \(on page \)](#) and [Cosmos React \(on page \)](#).

You can add primary fields to case types and data objects without the need to configure the UI; you only modify your data model and the system automatically adjusts the form. Pega Platform™ automatically generates a **Primary fields** view and adds it to the **Create**, **Edit**, and **Details** views.

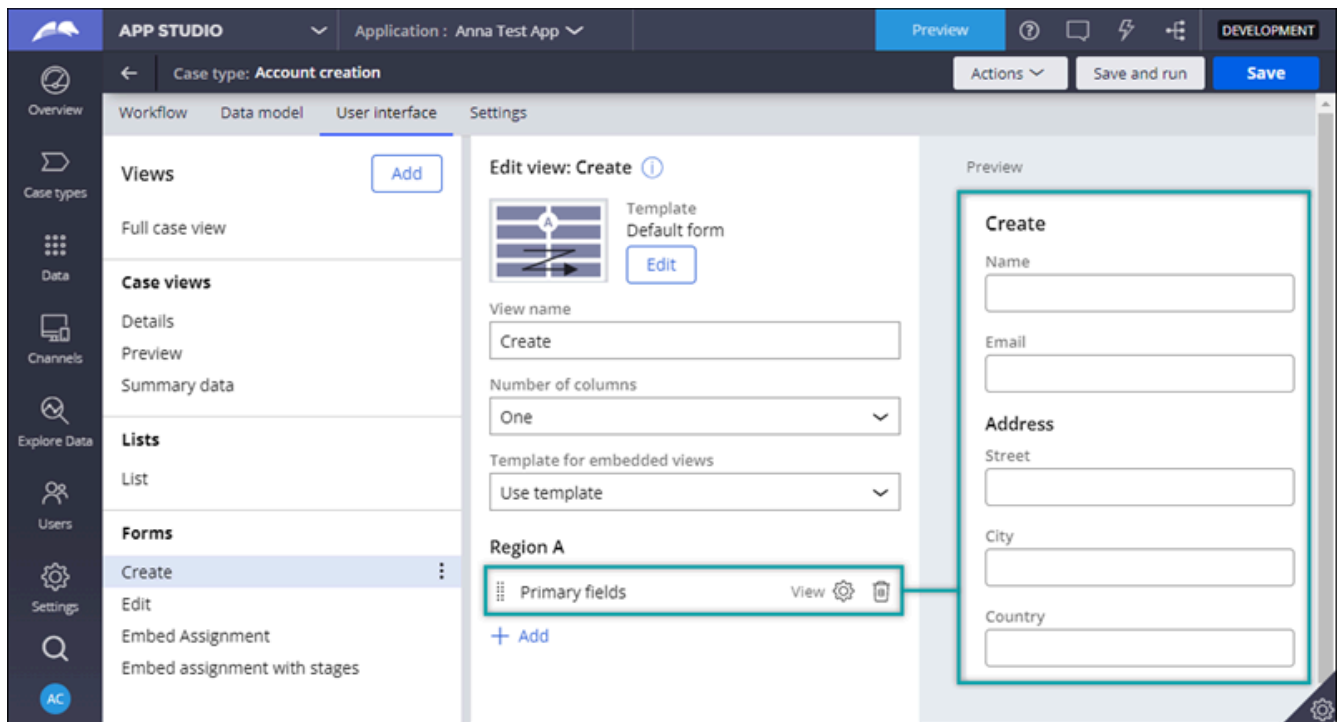


**Note:** Case types include the **Label** and **Description** primary fields by default.

For more information about views, see [Working with views \(on page \)](#).

For example, in a case type that captures customer information, you define **Name**, **Email**, and **Address** as primary fields. Then, you access the **Address** data object, and define its primary fields, such as **Street**, **City**, and **Country**. The system adds the fields automatically in the **Primary fields** view, as shown in the following figure:





Sample primary fields preview

1. In the navigation pane of App Studio, navigate to the item for which you want to define primary fields:

#### Choices

**Case types**

**Data objects**

#### Actions

Click **Case types**, and then click the case type that you want to open.

Click **Data**, and then click the data object that you want to open.

2. On the **Data model** tab, click **Primary fields**.
3. In the **Primary field configuration** modal dialog box, click **Add**, and then select the existing fields that you want to make primary.
4. **Optional:** To delete existing primary fields, click the **Remove from primary fields** icon.
5. **Optional:** To reorder the primary fields, drag and drop the fields in the appropriate positions.
6. Click **Submit**.

Adding fields (on page )

## Embedding data in a case

Manage your resources more efficiently by embedding data and connecting information directly to a case. Consequently, you avoid sourcing data from any other object. Embedding relevant data directly in a case if

the case is the only object that uses this piece of information helps you to organize data in your application in a more transparent way. For example, you can create an `Address` field and embed that field in a case type if this case type is the only element in your application that uses the information from the `Address` field.

To embed data in a case, you create a data field that stores the relevant information. You can add a data field to the following interfaces:

- Your case type data model. For more information, see [Configuring a data model for a case \(on page 150\)](#).
- A form in a view for a step in your case type for applications that do not use Cosmos React. For more information, see [Adding single-value fields to forms \(on page 174\)](#).
- Your application visual data model. For more information, see [Viewing an application data model \(on page 174\)](#).
- A data model of a data object. For more information, see [Viewing the data model for a data object \(on page 174\)](#).

1. Navigate to a place where you want to add a field:

Choices	Actions
<b>Add a field to a case type data model</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to a form in a view of a non-Cosmos React application</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>On the <b>Workflow</b> tab, in the <b>Case life cycle</b> section, click an assignment or an approval step.</li> <li>In the <b>Step</b> properties pane, click <b>Configure view</b>.</li> <li>In the dialog box, on the <b>Fields</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to an application visual data model</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Data</b>.</li> <li>In the <b>Data model</b> section, click <b>View</b>.</li> <li>In the data model, click a data object for which you want to create a data field.</li> <li>In the list of fields, click <b>Create new field</b>.</li> </ol>
<b>Add a field to a data object</b>	<ol style="list-style-type: none"> <li>In the navigation pane of App Studio, click <b>Data</b>.</li> <li>In the <b>Data objects</b> column, click the data object that you want to open.</li> <li>On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>



2. In the field configuration dialog box, for the **Field name** parameter, enter a unique label for the field.
3. In the **Type** list, select **Embedded data**.
4. In the **Data object** list, define a data object to store data:

Choices	Actions
<b>Create a new data object</b>	<ol style="list-style-type: none"> <li>a. In the list, select <b>Create a new data object</b>.</li> <li>b. In the <b>Data object name</b> field, enter a descriptive label for the data object.</li> <li>c. Click <b>OK</b>.</li> </ol>
<b>Reuse a data object</b>	<p>In the list, select a data object that stores the relevant data that you want to embed in the case type.</p> <p>The list contains all data objects from your application.</p>

5. In the **Options** list, define how you want to store the fields from the data object:
  - To create one entry for all the fields from the data object, select **Single record**.
  - To create a list of separate entries for the fields from the data object, select **List of records**.
6. **Optional:** To provide additional information about your data field, expand and complete the **Advanced** section:
  - To provide a different ID than the autogenerated ID, in the **ID** field, enter a new value.
  - To provide additional information about the field, in the **Description** text box, enter additional details.
7. Save your data field:
  - To save the field and instantly define another field, click **Submit & add another**.
  - To save your field and return to the configuration view, click **Submit**.
8. Click **Save**.

---

[Creating fields for capturing data \(on page 156\)](#)

[Cosmos React \(on page \)](#)

[Data modeling \(on page \)](#)

[Data objects overview \(on page \)](#)

## Referencing a data page

Provide data for your case types from integrations in your application by querying a data page. When you source data for your case directly from a data page, you reuse your resources in a transparent and



prescriptive way. For example, you can source information from a data page that has integration with a third-party system to display weather forecast in your application.

To reuse data from a data page in your case type, you create a data field that queries a data page. You can add a data field to the following interfaces:

- Your case type data model. For more information, see [Configuring a data model for a case \(on page 150\)](#).
- A form in a view for a step in your case type for applications that do not use Cosmos React. For more information, see [Adding single-value fields to forms \(on page 174\)](#).
- Your application visual data model. For more information, see [Viewing an application data model \(on page 174\)](#).
- A data model of a data object. For more information, see [Viewing the data model for a data object \(on page 174\)](#).

By creating a field that references a data page you can reuse information that a data page stores, in interfaces that typically do not allow referencing data pages, for example a condition builder or a decision table. As a result, you promote reuse of information across your application, save application development resources, and provide more flexible software. When you reference a data page, you receive entire content of the data page. If a data has parameters, you can specify the parameters to receive precise information that your business process requires. For example, when you source information about a weather forecast, you can use a location as a parameter so that your application displays the forecast for a selected city.

1. Navigate to a place where you want to add a field:

Choices	Actions
<b>Add a field to a case type data model</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to a form in a view of a non-Cosmos React application</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. On the <b>Workflow</b> tab, in the <b>Case life cycle</b> section, click an assignment or an approval step.</li> <li>c. In the <b>Step</b> properties pane, click <b>Configure view</b>.</li> <li>d. In the dialog box, on the <b>Fields</b> tab, click <b>Add field</b>.</li> </ol>
<b>Add a field to an application visual data model</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Data</b>.</li> <li>b. In the <b>Data model</b> section, click <b>View</b>.</li> </ol>

Choices	Actions
	<ul style="list-style-type: none"> <li>c. In the data model, click a data object for which you want to create a data field.</li> <li>d. In the list of fields, click <b>Create new field</b>.</li> </ul>
<b>Add a field to a data object</b>	<ul style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Data</b>.</li> <li>b. In the <b>Data objects</b> column, click the data object that you want to open.</li> <li>c. On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ul>

2. In the field configuration dialog box, for the **Field name** parameter, enter a unique name, for example `Weather forecast`.
3. In the **Type** list, select **Query**.
4. In the **Options** list, define how you want to reuse the fields from the source:
  - To create one entry for all the fields from the source, select **Single record**.
  - To create a list of separate entries for the fields from the source, select **List of records**.
5. In the **Data page** list, select the data page that you want to use as the source.
6. If the source data page has parameters, in the **Parameters** section, provide information about the parameters of the source data page:
  - To provide a static value to search for specific information in the data page, in the **pyID** list, select **Constant**, and then in the text field, provide the value that you want to use for searching the data page.
  - To select a field in the data page to source information, in the **pyID** list, select **Field**, and then, in the list, select the field that you want to use to source data.
7. **Optional:** To provide additional information about your data field, expand and complete the **Advanced** section:
  - To provide a different ID than the autogenerated ID, in the **ID** field, enter a new value.
  - To provide additional information about the field, in the **Description** text box, enter additional details.
8. **Optional:** To copy the referenced data into a case at run time, expand the **Advanced** section, and then select the **Reference data is copied to the case** check box.  
Copying data creates a snapshot of the referenced data and preserves the data in the target unchanged if data in the source changes.
9. Save your data field:
  - To save the field and instantly define another field, click **Submit & add another**.
  - To save your field and return to the configuration view, click **Submit**.
10. For fields that you add to a form, in the **Options** list, define the display mode for the field:

- To indicate that users can optionally complete the fields, select **Optional**.
- To indicate that users need to provide a value in the fields, select **Required**.
- To indicate that users can only view the fields, select **Read-only**.

11. Click **Save**.

---

Data page definition (*on page* )

Data modeling (*on page* )

[Creating fields for capturing data \(\*on page 156\*\)](#)

Cosmos React (*on page* )

[Embedding data in a case \(\*on page 159\*\)](#)

## Validating case data

Help users enter data values in a valid format by preventing processing errors when the users create or save a case. To save time, add validation on a case-wide level instead of validating recurring fields on each individual assignment. For example, in a Hire a candidate case type, you design a user view that includes a Hire date field. To ensure that users enter the date in a valid format, you add a validation rule for the Hire date field. However, if you need to configure the same Hire date field in multiple assignments throughout the case, you can add one validation rule for that field on a case-wide level that applies to all the Hire date fields in the case.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Settings** tab, click **Validation**.
4. Click **Configure** next to the action for which you want your application to validate user input.

Validation upon case creation is possible only for cases without the Create stage. If your case type includes the Create stage, and you want to configure validation upon case creation, configure validation on the next stage after the Create stage.

For more information, see [The Create stage \(\*on page 48\*\)](#) and [Defining the entrance criteria for a stage \(\*on page 58\*\)](#).

5. Click **Configure in Dev Studio** next to the action for which you want your application to validate user input.

Validation upon case creation is possible only for cases without the Create stage. If your case type includes the Create stage, and you want to configure validation upon case creation, configure validation on the next stage after the Create stage.



For more information, see [The Create stage \(on page 48\)](#) and [Defining the entrance criteria for a stage \(on page 58\)](#).

6. On the Create Validate form, in the **Context** section, in the **Add to ruleset** field, enter the name and version of an unlocked ruleset.

The default values in the **Label** and **Apply to** fields must remain unchanged, so that the rule resolution can identify the appropriate rules.

7. Click **Create and open**.

If you configure validation to occur when the user creates a case, the system considers the **Edit Validate: OnAdd** rule form. If you configure validation to occur when the user saves a case, the system considers the **Edit Validate: Validate** rule form. You complete the rule forms in the same way.

8. On the **Validate** tab, in the **Property** field, press the Down arrow key, and then select the property that you want to validate.
9. In the **Conditions** section, click **Add** to set a validation condition:
  - a. In the **Validation conditions** dialog box, in the **Select a function** field, press the Down arrow key, and then select a function to validate your property.
  - b. In the entry fields that display after you select a function, specify the parameters for your selection.
  - c. **Optional:** To display a custom message when the input value fails validation, enter the text in the **Message** field.
  - d. **Optional:** To enable execution of the conditions defined for the property, and the addition of multiple validation conditions, select the **Enable conditions** check box.  
If you clear the **Enable conditions** check box, you disable the execution of all of the conditions defined for the property, as well as the addition of multiple validation conditions.
  - e. **Optional:** To create a condition with multiple validation parameters, click **Add a row**, repeat substeps [9.a \(on page 165\)](#) through [9.b \(on page 165\)](#), and then add a logical operator to the condition.
  - f. **Optional:** To require the user at run time to enter a non-blank value for the corresponding property, select the **Required** check box.  
If you select the **Required** check box and the user leaves the field blank at run time, the system displays the `Value cannot be blank` message.
  - g. **Optional:** To have the system continue evaluating subsequent properties, even if the current evaluation fails, select the **Continue validation** check box.

If you enable the check box for multiple properties, the user can view all potential validation errors on one page.

h. Click **Submit**.

10. Click **Save**.

---

About Validate rules (on page )

## Calculating values with decision tables

To deliver an application that automatically at run time provides the most appropriate solutions, create logic with decision tables. Decision tables test a group of properties to match specified conditions and ultimately improve the flexibility of your application. By creating decision tables by using the low-code App Studio environment, you deliver software that can respond dynamically to varied conditions in business processes.

For example, you can create a decision table that calculates the account type for a customer based on a declared monthly income. The decision table returns a Standard account type for the income between 2,000 and 4,000 dollars, a Premium account type if the income is between 4,001 and 8,000 dollars, and a Gold account type for customers with income between 8,001 and 12,000 dollars.

 **Note:** Only applications that are based on Cosmos React support decision tables in App Studio.

Navigate to the calculated field:

1. In the navigation pane of App Studio, click **Data**.
2. In the **Data model** section, click **View**.
3. Open the calculated field:
  - To open a field from a case type data model, in the **Case types** section, click the case type that includes the field that you want to edit.
  - To open a field from a stand-alone data object, in the **Data objects** section, click the data object that includes the field that you want to edit.
4. In the **Fields** list, click the **Properties** icon in the row of the calculated field that you want to edit.

Associate the field with a decision table

5. In the **Configure field** dialog box, expand the **Advanced** section, and then, in the **Calculation** section, select **Use decision table**.
6. Define a decision table that you want to use:



- To create a new decision table, in the list of options, select **Custom decision table**, click the **Properties** icon, and then define the table.

For more information about creating decision tables for calculated fields, see [Authoring decision tables in App Studio \(on page 167\)](#).

- To reuse a decision table, in the list of options, select **Existing decision table**, and then, in the list of decision tables, select the table that you want to use.

7. Close the field configuration dialog box by clicking **Submit**.

### Related information

Building logic and calculating values in your application (on page )

Decision tables (on page )

[Configuring a data model for a case \(on page 150\)](#)

## Authoring decision tables in App Studio

Respond to dynamically changing business situations by calculating values in your business processes automatically. With decision tables, you can define conditions that include multiple elements and result in different outcomes. Consequently, you deliver an application that helps users successfully resolve business processes in various and dynamic scenarios. For example, you can create a decision table that calculates the type of account based on the monthly income of a customer. You can add additional variables to the calculation, such as the loan or mortgage history of the customer. To ensure that a business process can reach its goal even in an unexpected scenario, you define a value that the decision table returns if all of the calculations evaluate to false.

When you create a decision table, you select a field that you want to compare against specific values, and then you define a method of comparison. You can compare one field against multiple values to find the result that best fits your business needs. Each column includes a field to evaluate, and each row includes values that an application uses to compare against the field. At run time, the application evaluates a decision table from top to bottom, starting with the row at the top of the table.

1. Navigate to the place in your application where you want to use a decision table.
2. In the **Evaluate** dialog box for a calculated field, click **Add column**.
3. In the **Column** list, select the field that you want to evaluate at run time.
4. In the **Comparator** list, select a comparator to use for evaluation.  
A list of comparators depends on the type of the calculated field that you want to edit.
5. **Optional:** To add more fields to compare at run time, repeat steps 2 (on page 167) through 4 (on page 167).
6. Click **Submit**.



7. Click a cell in the column that you want to edit, and then enter a value to compare against the field at run time.  
If you have multiple columns, provide values for every column.
8. In the **Return** column, click a cell, and then enter a value that the decision table returns if the condition in the row that you want to edit evaluates to true.
9. **Optional:** To provide more values against the field to evaluate at run time, click the **Add** icon, and then repeat steps [7 \(on page 168\)](#) through [8 \(on page 168\)](#).
10. In the **Otherwise** row, in the **Return** column, click the cell, and then enter a value that the decision table returns if none of the rows evaluates to true.
11. **Optional:** To reuse the configured logic in the future, save the decision table:
  - a. Above the header of the decision table, click **Add to decision table library**.
  - b. **Optional:** To provide your custom name for the decision table, in the **Add to library for reuse** dialog box, in the **Name** field, enter the label for the decision table.  
By default, the system uses `Evaluate calculated field name` as the name for a decision table.
  - c. Click **Submit**.
12. Close the dialog box with decision table configurations by clicking **Submit**.

### Related information

Building logic and calculating values in your application ([on page 168](#))

Decision tables ([on page 168](#))

[Configuring a data model for a case \(on page 150\)](#)

### Views for cases

Views in cases represent forms that users interact with to provide information for case processing. When you create views, you decide what fields the users see and how they interact with the fields as the case moves towards a resolution. When you create, configure, and organize relevant fields in a form, you can clearly communicate to users data that they need to provide to process an assignment in a case.

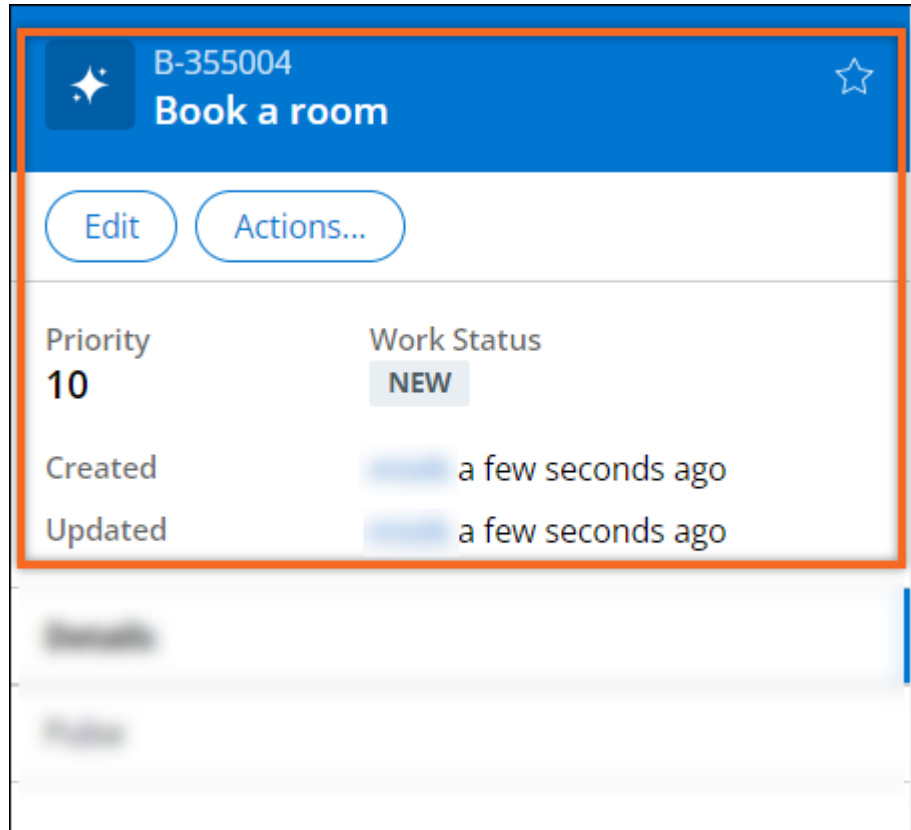
To save time, you can reuse dynamic views, which are standard forms that your application automatically manages for each case type that you define. By default, a dynamic view displays fields in a contextual data model during various events, such as case creation and resolution. You can edit a dynamic view to ensure that your case meets your specific business needs. You can reuse the following default dynamic views:

#### Case Summary (React UI)





Presents case information in a summary pane on the left side of the screen.



The screenshot shows a summary pane for a case titled "Book a room" with ID "B-355004". The pane includes an "Edit" button and an "Actions..." button. Below these are two columns of information: "Priority" (10) and "Work Status" (NEW). The "Created" and "Updated" timestamps are both listed as "a few seconds ago".

Priority	Work Status
10	NEW
Created	a few seconds ago
Updated	a few seconds ago

*Case summary*

**Create**

Prompts a user to provide information about a case, before processing the first step in the case life cycle.

**New Book your stay**

Arrival \*

Departure \*

Room type \*

Select...

Visitors \*

Cancel Create

Create view in Theme Cosmos

### Details (React UI)

Displays supplementary case information on the case summary pane.

✓ Create

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

Urgency 10.00

Label Book a room

Arrival 11/03/2021

Departure 11/10/2021

Room type Basic

Visitors 2

Meals Half board

Case details

### Edit

Provides controls for modifying case information at run time. Applications display this view with the **Edit details** label.

## Review

Presents case information to a user in a read-only format. Applications display this view with the **Case information** label.

To increase the flexibility and efficiency of your application, you can create your own custom views, and then associate them with steps in your case type. For example, in your case view, you can include fields that are read-only, such as a case ID, and fields that a user fills out, such as personal details or a phone number.

Provide views for your case types by completing the following actions:

[Skipping forms to collect information before case processing \(on page 172\)](#)

[Hybrid mode](#)

[Working with views \(on page \)](#)

## Configuring dynamic views

You can configure dynamic views that your application displays when users proceed through a case. By changing the default fields in these views, you can customize the forms that users access when they create, update, or review the case.



**Note:** Only standard Pega Platform applications support dynamic views. For information about configuring views for Cosmos React applications, see [Working with views \(on page \)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click the **Views** tab, and then click a dynamic view.  
Dynamic views display the **Dynamic view** icon.
4. In the dynamic view window, click **Edit this view**.
5. In the dialog window, update the dynamic view:
  - To add a preconfigured field, in the properties pane, expand the **Fields** section, and then click **Add to view** next to the field that you want to add.
  - To reuse an existing view, in the properties pane, expand the **Views** section, and then click **Add to view** next to the view that you want to add.

- To include a data object in your view, in the properties pane, expand the **Data objects** section, and then click **Add to view** next to the data object that you want to add.
- To create a new field, in the working area, click **Add field**, and then provide a descriptive name for the field.

6. Click **Submit**.

7. Click **Save**.

When users see a dynamic view in a case, the view contains the fields that you configure.

[Views for cases \(on page 168\)](#)

[Creating views for case types \(on page 173\)](#)

## Skipping forms to collect information before case processing

Speed up case creation and provide flexibility for your cases by skipping the Create dynamic view. When you need to capture user input as a part of case processing, you can skip the Create dynamic view. In that scenario, the case enters the first stage immediately, while users provide information during case processing. Then you can create the form to collect information from users to precisely meet your specific needs.

The Create dynamic view displays all fields that your application associates with the case data model. The application starts processing a case after users, such as customers, complete the fields in the Create dynamic view. When you need to display the fields in two columns, or when the data model contains fields that only customer service representatives use, you can customize the Create dynamic view.



**Note:** You can skip the Create view only in case types without the Create stage. For more information, see [The Create stage \(on page 48\)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click the **Settings** tab.
4. In the settings properties panel, click **General**.
5. In the **Behavior** section, select the **Skip 'Create' view when users create a new case** check box.
6. Click **Save**.

The next time that you create a case type instance, the case skips the Create form and moves directly to the first stage in the case life cycle.

[Views for cases \(on page 168\)](#)

[Working with Dev Studio forms \(on page \)](#)

## Creating views for case types

Create personalized views for your case types to capture the data that the case requires to resolve your business processes. As a result, you improve the flexibility of your application and accelerate case resolution, because you can define specific fields that collect information for specific cases. By creating a view, you can also lower application development time and costs, because you can reuse views for multiple case types. For example, you can create a view that captures feedback from a customer after resolving a case, such as a loan request, and then save time by reusing the view for other case types in which you need to collect feedback from a customer.

You can create views that are based on a table and display content in columns, or as forms that display a list of fields.

**Note:** The following procedure describes creating views for Theme Cosmos and classic Pega

**i** Platform applications. For information about creating views for Cosmos React applications, see [Creating views in Cosmos React \(on page \)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click **Views**.
4. Click **Create new view**.
5. In the **Create new** list, select the type of view that you want to create:
  - To create a view based on a table, click **List view**.
  - To create a view based on a form, click **Form view**.
6. Name your view:
  - To name a list view, in the **View name** field, enter a descriptive name.
  - To name a form view, in the **New view** field in the header of the configuration dialog box, enter a descriptive name.
7. Click **Submit**.

- UI components in a view (*on page* )
- [Adding single-value fields to forms \(\*on page 174\*\)](#)
- [Adding tables to views \(\*on page 178\*\)](#)
- [Reusing fields on forms \(\*on page 179\*\)](#)
- [Reusing forms \(\*on page 180\*\)](#)

## Adding single-value fields to forms

Collect specific, single pieces of information from users when they process a case by adding a single-value field to a form. For example, you can add a field that references a phone number to a form that prompts users to enter their personal and address details.

**Note:** The following procedure applies to Theme Cosmos and classic Pega Platform applications.

- ① For information about adding fields to forms in Cosmos React applications, see [Working with views \(\*on page\* \)](#) and [Configuring fields \(\*on page\* \)](#).

1. Navigate to a place where you want to add a single-value field:

Choices	Actions
<b>Add a single-value field to a case type</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. On the <b>Workflow</b> tab, click <b>Life cycle</b>.</li> <li>c. In the <b>Case life cycle</b> section, click an assignment or an approval step.</li> <li>d. In the <b>Step</b> properties panel, click <b>Configure view</b>.</li> <li>e. In the dialog box, click <b>Add field</b>.</li> </ol>
<b>Add a single-value field to a data object</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Data</b>.</li> <li>b. Click the data object that you want to open.</li> <li>c. On the <b>Data model</b> tab, click <b>Add field</b>.</li> </ol>

2. In the dialog box, in the **Field name** field, enter a unique name for the field.
3. In the **Type** list, define a field type:

## Choices

### Add a text paragraph to the form

## Actions

- In the **Type** list, select **Text (paragraph)**.
- Click the **Configure paragraph** icon.
- In the dialog window, in the **Display as** list, define whether to display the paragraph as plain or rich text.
- Click **Submit**.

The following figure shows a plain text paragraph at run time in a Theme Cosmos application:

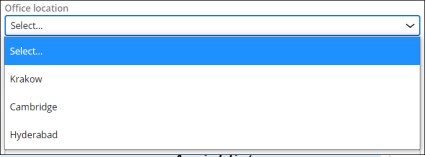
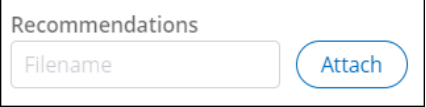
Describe your symptoms  
Plain text paragraph

### Add a picklist to the form


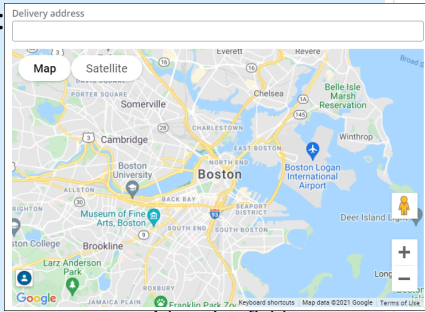
- In the **Type** list, select **Picklist**.
- Click the **configure options** icon.
- In the dialog window, in the **Display as** list, define a display mode for the picklist.
- In the **Picklist** options list, define choices for the picklist.
  - To create your own choices, select **Local**, and then click **Add choice**.
  - To source the choices from a data page, select **Data page**, and select a data page from the list.

Then, choose the identifier field and the display field.
- Click **Submit**.

The following figure shows a picklist configured as a

Choices	Actions
	<p>drop-down list at run time in a Theme Cosmos application:</p>  <p><i>A picklist</i></p>
<p><b>Add an attachment field to the form</b></p>	<ol style="list-style-type: none"> <li>In the <b>Type</b> list, select <b>Attachment</b>.</li> <li>Click the <b>Choose the attachment category</b> icon.</li> <li>In dialog window, in the <b>Attachment category list</b>, select the category.</li> <li>Click <b>Submit</b>.</li> </ol> <p>The following figure shows an attachment field at run time in a Theme Cosmos application:</p>  <p><i>An attachment field</i></p>
<p><b>Add a user reference to the form</b></p>	<ol style="list-style-type: none"> <li>In the <b>Type</b> list, select <b>User reference</b>.</li> <li>Click the <b>Configure user reference</b> icon.</li> <li>In the dialog window, in the <b>Select record using</b> list, define whether users can search for a user ID by using a search box or a drop-down list.</li> <li>Click <b>Submit</b>.</li> </ol> <p>The following figure shows a user reference field at run time in a Theme Cosmos application. When users start typing or press the Down</p>



Choices	Actions
<p><b>Add a location field to the form</b></p>	<p>arrow key, an application prompts relevant user references.</p>  <p><i>A user reference field</i></p>
	<p>a. In the <b>Type</b> list, select <b>Location</b>.</p> <p>b. Click <b>Submit</b>.</p> <p>The following figure shows a location field at run time in a Theme Cosmos application:</p>  <p><i>A location field</i></p>

4. If you are adding fields to a case type in App Studio or Dev Studio, in the **Options** list, define a display mode for the field:
  - To indicate that users can optionally complete the field, select **Optional**.
  - To indicate that users need to provide the value in the field, select **Required**.
  - To indicate that users can only view the field, select **Read-only**.
  - To indicate that users can only view the field and an application determines the value at run time, select **Calculated (read-only)**.
5. Click **Submit**.

[Creating a calculated field \(on page 181\)](#)

[Creating views for case types \(on page 173\)](#)

[Property type changes and ruleset versions \(on page \)](#)

[Restricting user input in a field \(on page 181\)](#)

[Defining the fields in a data object \(on page \)](#)

## Adding tables to views

Help users access and compare data by setting up a form with a table. Tables are a basic component that you can use to create a clear interface in information-heavy contexts. For example, tables that display the status of open cases can help managers quickly assess how a project is progressing.

**Note:** The following procedure applies to Theme Cosmos applications. For information about configuring views in Cosmos React applications, see [Creating views in Cosmos React](#) (on page ).

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the case working area, click the **Views** tab.
3. Click **Create new view > List view**.
4. In the **View name** field, provide the name under which the view appears on the view list.
5. In the **Data page** field, select the list from which you want to source the table contents.
6. In the **Display name** field, enter the text that appears in over the view in the UI, and then define how you want to present the data to the user:

Choices	Actions
<b>Table</b>	In the <b>Template</b> list, select <b>Table</b> .
<b>Tile-based gallery</b>	<ol style="list-style-type: none"> <li>a. In the <b>Template</b> list, select <b>Gallery</b>.</li> <li>b. In the <b>Card header</b> list, select the field that you want to use as the header for your tile.</li> <li>c. In the <b>Secondary text</b> field, select the field that you want to use as the source of additional text on your tile.</li> </ol>

7. In the **Columns** section, add the columns that you want to include in your table by clicking **Add**, and then select the property that you want to associate with the column.  
The system sources columns from the current case type.
8. In the **Column to take up remaining width**, define which column stretches to fill free space in your table.
9. **Optional:** To define what data is displayed in the table, set filters:

- a. In the **Filter by** list, select **Custom**, and then click the **Properties** icon.
  - b. In the **Condition Builder** window, define the expression that you want to use to filter table contents.
  - c. **Optional:** To build a complex filter with more than one expression, click the **Add** icon, and then define the additional expression.
10. Define how the application sorts data in the table:
- a. In the **Sort by** list, add the property by which you want sort the list by clicking **Add**, and then selecting the field that you want to use as a filter.
  - b. In the list that appears next to the property field, select the order in which you want to sort the table.
11. **Optional:** To enable bulk processing of list items, select the **Allow bulk actions** checkbox.



**Tip:** Bulk processing is available for tables on case pages and landing pages. The actions in the bulk action menu are case-wide actions.

12. **Optional:** To define additional table behavior, select the checkboxes that represent specific personalization settings.
13. Click **Submit**.

## Reusing fields on forms

Save time and build forms that are convenient to maintain by referencing fields on other, existing forms. For example, you can reuse a field group to capture different user details on a new form, such as name, surname, mailing address and phone number, instead of creating separate fields for each item.

**Note:** The following procedure applies to Theme Cosmos and classic Pega Platform applications.



For information about adding fields to forms in Cosmos React applications, see [Configuring details views \(on page \)](#) and [Configuring fields \(on page \)](#).

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Workflow** tab, click **Life cycle**.
3. In the **Case life cycle** section, click an assignment or an approval step.
4. In the **Step** properties panel, click **Configure view**.

5. In the dialog window, click **Fields**.
6. Click **Add to view** next to the field that you want to reuse.
7. **Optional:** To change the default display mode of the field, in the **Options** list, select a new mode:



**Note:** You cannot change the display mode of a field when its value is calculated by using an expression.

- To indicate that users can optionally complete the field, select **Optional**.
- To indicate that users need to provide the value in the field, select **Required**.
- To indicate that users can only view the field, select **Read-only**.
- To indicate that users can only view the field and an application determines the value at run time, select **Calculated (read-only)**.

You configure the default display mode when you first create a field. For more information, see [Adding single-value fields to forms \(on page 174\)](#).

8. Click **Submit**.

---

[Configuring dynamic views \(on page 171\)](#)

[Reusing forms \(on page 180\)](#)

## Reusing forms

You can reduce development time and maintain consistent layouts between sets of fields in your case type by embedding forms in other forms. For example, you can prompt users to review their employment history by embedding their Work History form in a Confirmation form.

**Note:** The following procedure applies to Theme Cosmos and classic Pega Platform applications.



For information about adding fields to forms in Cosmos React applications, see [Configuring details views \(on page \)](#) and [Configuring fields \(on page \)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, click an assignment or an approval step.
5. In the **Step** properties panel, click **Configure view**.



6. In the dialog window, click **Views**.
7. Click **Add view** next to the view that you want to reuse.
8. **Optional:** To view or edit the form that you want to reuse, click **Open**.
9. In the **Options** list, choose an edit mode for the reused form:
  - To maintain the individual setting for each field on the form, select **Auto**.  
  
Fields can be read-only, required, or optional.
  - To display all the fields as read only, select **Read-only**.
10. Click **Submit**.

---

Restrictions on field and form reuse (*on page* )

[Adding single-value fields to forms \(on page 174\)](#)

Supported display modes by field type (*on page* )

## Restricting user input in a field

Configure fields in your data model to define the way that they store or display information to help users provide data in a format that is required in your business scenario. As a result, you minimize the risk of providing erroneous data and improve quality of case processing. For example, you can create a calculated field that automatically sums up values in a case. Consequently, you increase a level of automation in your case type and accelerate processing of cases. You can also implement validations to ensure that users enter data in a correct format.

Explore the following articles to learn more about solutions that you can use in your case types:

---

Defining the fields in a data object (*on page* )

[Creating views for case types \(on page 173\)](#)

## Creating a calculated field

Save time and improve the accuracy of the information in your case processing by creating a calculated field that automatically returns a value based on the input data. For example, you can create a field that calculates a length of stay at a hotel based on arrival and departure dates that a user provides.

1. Navigate to a place where you want to add a calculated field:

Choices	Actions
<b>Add a calculated field to a case type</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. Click the <b>Data model</b> tab, and then click <b>Add field</b>.</li> </ol>
<b>Add a calculated field to a data object</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Data</b>.</li> <li>b. Click the data object that you want to open.</li> <li>c. Click the <b>Data model</b> tab, and then click <b>Add field</b>.</li> </ol>

2. In the **Add field** modal dialog box, in the **Field name** field, enter the name of the new field.
3. In the **Type** field, select a field type that supports calculations, for example, **Integer**.
4. Expand the **Advanced** section, and then select the **This is a calculated field (read-only)** checkbox. The checkbox is visible only for fields that support calculation.
5. Define how an application calculates the field value:

Choices	Actions
<b>Calculate a value by using an expression</b> <p><b>Note:</b> All types of calculated fields support expressions.</p>	<ol style="list-style-type: none"> <li>a. In the <b>Calculation</b> section, in the <b>Function</b> list, select <b>Use expression</b>.</li> <li>b. In the text field, enter an expression that you want to apply.</li> </ol>
<b>Calculate a value by using a custom decision table</b> <p><b>Note:</b> All types of calculated fields support decision tables.</p>	<ol style="list-style-type: none"> <li>a. In the <b>Calculation</b> section, in the <b>Function</b> list, select <b>Use decision table</b>.</li> <li>b. In the drop-down list, select <b>Custom decision table</b>, and then click the <b>Gear</b> icon.</li> <li>c. In a dialog box that opens, configure your decision table.</li> </ol>

Choices	Actions
	<p>For more information, see <a href="#">Calculating values with decision tables (on page 166)</a>.</p> <p>d. Click <b>Submit</b>.</p>
<p><b>Calculate a value by using an existing decision table</b></p> <div data-bbox="240 596 841 730" style="background-color: #e1f5fe; padding: 5px; border: 1px solid #00796b;"> <p><span style="font-size: 1.2em;">i</span> <b>Note: All types of calculated fields support decision tables.</b></p> </div>	<p>a. In the <b>Calculation</b> section, in the <b>Function</b> list, select <b>Use decision table</b>.</p> <p>b. In the drop-down list, select <b>Existing decision table</b>, and then select the decision table that you want to use.</p> <p>c. Click <b>Submit</b>.</p>
<p><b>Calculate a value of a currency, decimal, integer, or percentage field by using a function</b></p>	<p>a. In the <b>Calculation</b> section, in the <b>Function</b> list, select a type of a function that you want to apply.</p> <p>b. In the <b>Field</b> list, select a single-value field from an embedded data field in your data model.</p> <p>c. Click <b>Submit</b>.</p>

6. Click **Save**.

## Validating field values on a form

Ensure that users enter valid data on a form before the case moves to the next step in the life cycle. By validating field values, you prevent processing errors and speed up the case resolution process. If the data that the user enters on a form at run time meets the condition in a validate rule, the application displays an error message. For example, you can ensure that the customer provides their name and address before placing an order. If the customer leaves the fields empty, the application displays an error message.

1. In the navigation pane of Dev Studio, click **Case types**.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, select the step to which you want to apply the validation condition.
4. In the **Step** properties pane, on the **General** tab, click **Configure view**.
5. **Optional:** To ensure that the user enters a value in a field before submitting the form, set the field as **Required**:



Choices	Actions
<b>Mark a field as required in a standard Pega Platform application</b>	<ol style="list-style-type: none"> <li>In the view configuration window, on the <b>Fields</b> tab, locate the field that you want to set as <b>Required</b>.</li> <li>In the <b>Options</b> list, select <b>Required</b>.</li> <li>Click <b>Submit</b>. When the user tries to submit the form without a value in a required field, the application displays an error message.</li> </ol>
<b>Mark a field as required in a Cosmos React application</b>	For more information about defining field behavior in Cosmos React, see <a href="#">Configuring field behavior (on page 184)</a> .

- In the view configuration window, define a new validation condition:
  - In a standard Pega Platform application, click the **Validations** tab, and then click **Add condition**.
  - In a Cosmos React application, click the **Conditions** tab, and then click **Add condition**.
- Optional:** To display an error message when the validation fails, in the **Message** field, enter the text of the message.
- Define the validation conditions:

Choices	Actions
<b>Define a validation condition for a field</b>	<ol style="list-style-type: none"> <li>In the list of values, select <b>Fields</b>, and then select the name of the field that you want to validate.</li> <li>In the comparator list, select the test that you want to perform on the field.</li> <li>In the value field, enter or select a value to compare against the user input. The <b>Select values</b> option lists the fields and values that you can use in the condition.</li> </ol>



Choices	Actions
<b>Apply an existing when condition as a validation condition</b>	<ol style="list-style-type: none"> <li>In the list of values, select <b>When conditions</b>, and then select the name of the when condition that you want to apply.</li> <li>In the comparator list, select the test that you want to perform on the when condition.</li> </ol>

9. **Optional:** To create a condition with multiple validation parameters, add a logical operator to the condition:

Choices	Actions
<b>The condition passes when all properties meet the criteria</b>	<ol style="list-style-type: none"> <li>Click <b>Add a row</b>.</li> <li>In the operator list, select <b>and</b>.</li> <li>In the list of fields, select the name of the field that you want to validate or the name of the when condition that you want to apply.</li> <li>In the comparator list, select the test that you want to perform on the field or on the when condition.</li> <li>In the value field, enter or select a value to compare against the user input. The <i>Select values</i> option lists the fields and values that you can use in the condition.</li> </ol>
<b>The condition passes when any of the properties meet the criteria</b>	<ol style="list-style-type: none"> <li>Click <b>Add a row</b>.</li> <li>In the operator list, select <b>or</b>.</li> <li>In the list of fields, select the name of the field that you want to validate.</li> <li>In the comparator list, select the test that you want to perform on the field or on the when condition.</li> </ol>

Choices	Actions
	<p>e. In the value field, enter or select a value to compare against the user input. The <code>Select values</code> option lists the fields and values that you can use in the condition.</p>

10. **Optional:** To define more validation conditions, repeat steps 6 (on page 184) through 9 (on page 185).
11. Click **Submit**.

At run time, the application evaluates all validation conditions. If the user enters a value that meets any of the validation conditions, the respective error message appears. The user can submit the form after they correct the invalid fields.

Working with Dev Studio forms (on page )  
[Restricting user input in a field \(on page 181\)](#)

## Validating field input in complex scenarios

Create validation rules to check whether the data values that users enter meet specific criteria, and whether the system can process the information correctly.

While step validation supports relatively simple comparisons, validation rules help you create more complex operations that include functions and conditional processing. For example, a loan application might require different validation conditions depending on the risk rating of the applicant. You can set up a `RiskLevel` property with a value of *High*, *Medium*, and *Low*, and define a set of validation conditions for customers that represent *High*, *Medium* and *Low* risk levels.

**Note:** You define validation conditions that check whether the user input in a form is incorrect. If the user input matches the validation parameters, an error message appears when the user tries to submit the form.

1. In the header of Dev Studio, click **Create > Process > Validate**.
2. On the **New** tab, define the label, context, and ruleset for the new validation rule, and then click **Create and open**.
3. On the **Input** tab, define the input that the application uses to determine which set of validation conditions to apply at run time:



- To define only one validation set, select **None**.
  - To define a separate validation set for each value of a property, select **Input property**, and then select the target property.
  - To define a separate validation set for each case status, select **Proposed work status**.
  - To define a separate validation set for each flow action, select **Flow Action Name**.
  - To define a separate validation set for each stage, select **Stages**.
4. On the **Validate** tab, in the **Property** field, enter the property that you want to test.
  5. In the **Conditions** section, create a validation set by clicking **Add**.
  6. In the **Validation conditions** window, define the validation set:
    - a. Ensure that the **Enable conditions** checkbox is selected.
    - b. **Optional:** To indicate that the user must fill out a field, select the **Required** checkbox.
    - c. **Optional:** To force the system to evaluate subsequent properties even if the current evaluation fails, select the **Continue validation** checkbox.
    - d. In **Select a function**, choose the function that you want to use to evaluate the property.
    - e. In the fields below, enter the expression for the function.
    - f. In the **Message** field, enter the text you want to display when the condition is true.
    - g. **Optional:** To add an additional validation function, click **Add a row**, and then select a logical operator to link the functions.
    - h. Click **Submit**
  7. If your validation rule includes multiple validation sets for different input values, you can define additional validation sets:
    - a. In the upper right corner of the **Validate** tab, click **Add a new condition**.
    - b. In the **Input** row, enter the input value for which you want to create a validation set.
    - c. Configure the validation set by completing steps [6.b \(on page 187\)](#) through [6.h \(on page 187\)](#).
    - d. **Optional:** To run another validation set after the current validation completes, in the **Also execute** row, specify the target validation set.
  8. **Optional:** To run additional validation rules after the current validation completes, in the **Additional validation** row, select the target validation rule.
  9. Click **Save**.

---

### Related information

[Validating character patterns \(on page 188\)](#)

## Adding a validation rule to a flow action

Ensure that the data that your users provide meets the conditions that a case requires in order to go forward. By assigning validation rules to flow actions, you can prevent users from entering information that your application cannot process, and reduce the number of processing errors. For example, a loan



application might require the customer to provide a down payment. You create a validate rule that checks whether the down payment is sufficient. Next, you assign the rule to a flow action that executes at the end of the step in which the customer provides loan information. If the amount of the down payment is too low, the validate rule prevents the case from proceeding until the customer increases the down payment.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Workflow** tab, hover over the name of the process to which you want to add a validation rule, and then click **Configure process** icon.
3. In the top right corner of the screen, click **Open process**.
4. On the canvas, right-click the connector to which you want to add the validation rule, and then click **Open Flow Action**.
5. In the flow action rule, open the **Validation** tab.
6. In the **Validate** field, select the validation rule that you want to assign to the connector.
7. Click **Save**.

## Validating character patterns

Check whether the input that your user provides is relevant. By verifying character patterns with edit validate rules, you can ensure that the data that users enter into forms matches the field requirements, and reduce the number of processing errors. For example, if you want your users to enter a valid US postal code in a field, you can use an edit validate rule to ensure that the input is consistent with the ZIP Code format. Pega Platform includes a wide variety of ready-to-use edit validate rules that come with default error messages.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Validate**.
3. Open the validate rule that you want to edit.
4. In the **Conditions** section, open a validation set by clicking **Edit**.
5. In the **Validation conditions** window, click **Add a row**.
6. In the new row, in the **Select a function** field, select **invokeValidate**.
7. In the **Validation of** field, select the field that you want to check.
8. In the **using** field, select one of the available edit validate rules.
9. Click **Submit**.
10. Click **Save**.

## Remote case types

In organizations that use multiple applications, case workers often switch contexts to work. Remote case types help you provide tools for uninterrupted and more efficient case resolution. Case workers can work



from multiple applications within the context of one application without logging in and out numerous times a day.

With remote case types, in one application, case workers can perform work that comes from another application. For example, a customer service representative (CSR) who deals with financial operations of VIP customers logs in to the following applications:

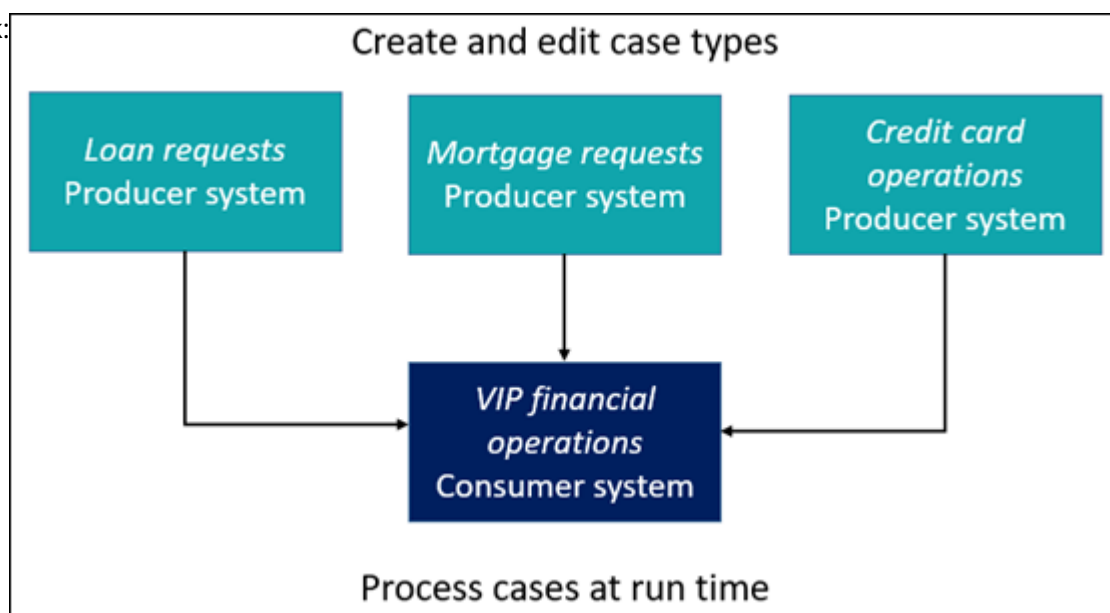
- Loan requests
- Mortgage requests
- Credit card operations

The CSR switches applications multiple times during a workday, which results in more time spent switching applications and less time performing work. Instead, you can create a VIP financial operations application that collects relevant cases from the other applications. Consequently, the CSR can perform the entire work in only one application.

## System architecture

For remote cases to work, a producer system hosts an application that has a case type that you want to process at run time. Meanwhile a consumer system hosts an application that you want to connect to the case type. In addition, you can configure one system as a producer and a consumer if you host multiple applications in one system. First, you create a case type in the producer system, and then create this case type as the remote case type in the consumer system. You populate your case types with actions and data model fields in the producer system, as the consumer system only references the case types. You cannot edit the case types in the consumer system. The following figure presents a sample use case that shows how remote case types

work:



A sample model of remote case types

the sample use case, in the VIP financial operations application, CSRs can process case types that come from other applications:

- Loan requests
- Mortgage requests
- Credit card operations

## Implementation considerations

Consider the following factors before you implement remote case types:

- Build both producer and consumer applications by using Cosmos React. Only applications that are based on Cosmos React support remote case types.
- Before you can work on a remote case type in a remote application, establish a secure connection between the producer and consumer systems.
- Ensure that you configure both producer and consumer systems by using the same operator ID.
- Ensure that you configure both producer and consumer systems to work with SSL certificates.
- After an update to a higher version of Pega Platform™, you must override the *Remotecasetype\_Consumer* token profile again. For more information, see [Configuring a consumer system for remote case types \(on page 194\)](#).
- After you import remote case types to a different environment, you must edit the connection details of each of your remote case types.

---

Cosmos React (on page )

Creating an operator ID (on page )

Managing work across your team (on page )

### Establishing trust between applications for remote case types

To ensure that remote case types data can pass between applications, establish secure connection between producer and consumer systems. By performing necessary authorization and security actions, you ensure that the producer system can send data and the consumer system can fetch data correctly.

### Configuring a producer system for remote case types

To pass remote case types to another application, configure the system that stores your case types as a producer system. As a result, you provide a framework for case workers to process work from multiple applications without changing context, which promotes efficiency and saves time in your organization. For example, your system might host Loan requests and Credit card operations applications. You want to publish case types from these applications to a VIP customers operations application so that users can

perform work by logging in to this application only. You configure the system with the Loan requests and Credit card operations applications as the producer system first.

Accessing remote case assets that are available in the producer system requires performing security configurations. Pega Platform provides the OAuth 2.0 client registration, identity mapping, and token profile that you copy into your ruleset and edit to meet your business needs. The security configuration also requires your own truststore.

---

Key management system for application data encryption (*on page* )

Cosmos React (*on page* )

Authentication (*on page* )

## Preparing OAuth 2.0 client registration for remote case types

To ensure that remote application from a consumer system can properly communicate with your producer system, prepare the OAuth 2.0 client registration in your producer system. As a result, you create a secure and reliable connection between your producer and consumer systems.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **OAuth 2.0 Client Registration**.
3. In the list of instances, open the **Remotecasetype\_Producer** OAuth 2.0 client registration.
4. In the form header, click **Save > Save as**.
5. On the **Save OAuth 2.0 Client Registration As** form, in the **Client Name** field, enter a unique label, and then click **Create and open**.
6. In the form header, in the **RS** section, click **Edit**, and then, in the text field, provide the ruleset to associate with your application.
7. In the **Client Credentials** section, download the client ID and client secret by clicking **View & download**, and then click **Save**.



**Note:** You provide the client ID and client secret credentials when you register a remote system during creation of a remote case type. Ensure that you store the credentials safely.

---

Authentication (*on page* )

Creating an authentication profile (*on page* )

Copying a rule or data instance (*on page* )

## Preparing a token profile for remote case types

After you prepare the OAuth 2.0 client registration for your remote case types, prepare a token profile. Token profiles secure data exchange between two parties, such as producer and consumer systems.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **Token Profile**.
3. In the list of instances, open the **Remotecasetype\_Producer** token profile.
4. In the form header, click **Save > Save as**.
5. On the **Save Token Profile As** form, in the **Name** field, provide a unique label, and then click **Create and open**.



**Note:** Make note of the token profile name as this name is required during preparing identity mapping.

6. In the form header, in the **RS** section, click **Edit**, and then, in the text field, provide the ruleset to associate with your application.
7. In the **Claims mapping** section, click **Add a row**.
8. In the **Claim name** field, enter `sub`, and then, in the **Property** field, enter `pxRequestor.pyUserIdentifier`.
9. In the **Claims mapping** section, click **Add a row**.
10. In the **Claim name** field, enter `exp`, and then, in the **Property** field, enter `OperatorID.pyExpirationDate`.

The following figure shows a correct configuration of the claims mapping:

Claim name	Property
sub	pxRequestor.pyUserIdentifier
exp	OperatorID.pyExpirationDate

Claims mapping for the producer system

11. Click **Save**.



Security operations (*on page* )  
 Creating token profiles (*on page* )  
 Copying a rule or data instance (*on page* )

## Preparing identity mapping for remote case types

After you prepare the client registration and token profile for your producer system, prepare identity mapping to ensure that Pega server correctly identifies operators in your application. You also define how to map the user identity information for use in the Pega application.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **Identity Mapping**.
3. In the list of instances, open the **Remotecasetype\_Producer** identity mapping.
4. In the form header, click **Save > Save as**.
5. On the **Save Identity Mapping As** form, in the **Name** field, enter a unique label, and then click **Create and open**.
6. In the form header, in the **RS** section, click **Edit**, and then, in the text field, provide the ruleset to associate with the identity mapping.
7. In the **Token validation** section, in the **Token processing profile** field, enter the name of the token profile that you created for the producer system, as in the following example:
8. Click **Save**.

Authentication (*on page* )  
 Authenticating requests in services (*on page* )  
 Creating an identity mapping data instance (*on page* )  
 Copying a rule or data instance (*on page* )

## Mapping the truststore to the token profile for remote case types

After you prepare the identity mapping, map the token profile to the truststore to ensure a secure encryption of data between your producer and consumer systems.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **Token Profile**.
3. In the list of instances, open the token profile that you created for your producer system.  
 For more information, see [Preparing a token profile for remote case types \(on page 192\)](#).
4. In the **Security** section, in the **Truststore** field, enter the truststore that you created for the producer system.
5. Click **Save**.



---

Encryption (*on page* )

Key management system for application data encryption (*on page* )

Keystores (*on page* )

## Mapping the identity mapping to the client registration for remote case types

Ensure that the producer and consumer system can communicate correctly and securely by mapping the identity mapping to the client registration that you created for the producer system.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **OAuth 2.0 Client Registration**.
3. In the list of instances, open the OAuth 2.0 client registration that you created for the producer system.  
For more information, see [Preparing OAuth 2.0 client registration for remote case types \(on page 191\)](#).
4. In the **Supported grant types** section, in the **Identity mapping** field, enter the identity mapping that you created for the producer system.  
For more information, see [Preparing identity mapping for remote case types \(on page 193\)](#).
5. Click **Save**.

---

Authentication (*on page* )

Authenticating requests in services (*on page* )

Creating an identity mapping data instance (*on page* )

Creating token profiles (*on page* )

## Configuring a consumer system for remote case types

To process cases that come from another application, configure the system that hosts your destination application as a consumer. As a result, at run time, users can perform work from multiple applications within the context of one consumer application, without the need to switch between different products. For example, a customer service representative (CSR) can use a Financial operations application to work on cases that belong to the Loan requests and Mortgage requests applications. In this case, the Financial operations application is on a consumer system, while the Loan requests and Mortgage requests applications operate on a producer system.

Configuring a consumer system requires a keystore and truststore that support connection to the producer system, as well as copying and then editing a token profile that Pega Platform includes by default.



1. Create a truststore with the `Remotecasetype_Truststore` name, and then upload the `.jks` truststore file of the producer system.  
For more information, see [Creating a keystore for application data encryption \(on page 195\)](#).
2. Create a keystore with the `Remotecasetype_Keystore` name, and then upload the `.jks` keystore file of the consumer system.  
For more information, see [Creating a keystore for application data encryption \(on page 195\)](#).
3. In the navigation pane of Dev Studio, click **Records**.
4. Expand the **Security** category, and then click **Token Profile**.
5. In the list of instances, open the **Remotecasetype\_Consumer** token profile.

**Note:**

- For remote case types to work correctly, configure the token profile that Pega Platform includes by default instead of copying or creating a new instance of the token profile.
- After an update to a higher version of Pega Platform, you must override the `Remotecasetype_Consumer` token profile again.

6. In the **Security** section of the token profile, configure security settings:
  - a. In the **Security configuration** list, select **Signature**.
  - b. In the **Signature type** list, select **Asymmetric**.
  - c. In the **Signature type algorithm** field, press the Down arrow key, and then select **RS256**.
  - d. In the **Keystore** field, enter the keystore that you created in step 2 ([on page 195](#)).



**Tip:** The JSON web key set URI from the **Security** section of this token profile is the URL that you use when you create a truststore for the producer system.

When you select the keystore, the system autopopulates the alias value.

- e. In the **Password** field, enter the password related to the keystore.
7. Click **Save**.

Key management system for application data encryption (*on page* )  
 Cosmos React (*on page* )  
 Authentication (*on page* )

## Creating remote case types

Provide tools for fast and seamless work processing for organizations that use multiple applications by creating remote case types. With remote case types, users can process work from one application within another application, without changing the context and logging in to many different products. For example, a customer service representative (CSR) who processes loan requests and mortgage requests can process work by using only one application, even if mortgage and loan requests come from two different applications. To provide this option for a CSR, in a Loan operations application, you create a Loan request case type, and then associate the case type with the Mortgage operations remote application. As a result, the CSR can create and process loan request cases from the Mortgage operations application.



**Note:** When you create a remote case type, the system creates a data page that fetches a list of cases of this case type. The data page name includes the appendix `List`. For example, if you create a `Review loan request` case type, the system creates the `D_ReviewLoanRequestList` data page. Do not delete this data page in the producer system.

1. In the navigation pane of App Studio, click **Case types**.
2. In the **Case types** header, click **New**.
3. In the **Create case type** dialog box, enter the case type details:
  - a. In the **Case type name** field, enter a label for your case type, for example, `Loan requests`.
  - b. Expand the **Advanced** section, and then, in the **Type** list, select **Remote**.
  - c. Click **Next**.
4. Connect the case type to a remote application:

### Choices


**Use an existing connection**

**Create a new connection with a remote application**

### Actions

In the **Connect to remote application** dialog box, in the **Remote application** list, select the application that you want to connect to the case type, and then click **Connect**.

- a. In the **Connect to remote application** dialog box, in the **Remote application** list, select **Create new remote application**.

Choices	Actions
	<p>b. In the <b>Name</b> field, enter the name of your remote application, for example, <code>Mortgage operations</code>.</p> <div data-bbox="938 415 1468 695" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 10px;"> <p><b>Note:</b> This application already exists in your producer system.</p> <p> You create a connection between your current application and the producer application.</p> </div> <p>c. In the <b>System URL</b> field, enter the URL of the producer system. For multitenant systems, ensure that you include the tenant hash in the URL. For other systems, ensure that you include the context root in the URL.</p> <p>d. In the <b>Client ID</b> and <b>Client secret</b> fields, enter the credentials for connecting the case type to the application. You generate the credentials when you configure the producer system.</p> <p>e. Click <b>Connect</b>.</p>

5. In the **Select case type** dialog box, select a case type to source from the producer system:

- a. In the **Application** list, select the application that stores the case type.  
If your producer system has only one application that is configured to support remote case types, the system autopopulates the **Application** field. The list displays only those applications that you can access in the producer system.
- b. In the **Case type** list, select the case type that you want to source from the producer application.
- c. Click **Next**.

6. **Optional:** If you create a new connection with a remote application, to override default configuration for your application in other environments, such as staging production, provide new connection details:
  - a. In the **Environment configuration** dialog box, select the **Edit connection details in other environments** check box.
  - b. Provide connection details as necessary.
  - c. Click **Next**.
7. Import the data model from the producer system:
  - a. On the **Data model** tab of the case type from the producer system, click **Add fields**.
  - b. In the **Add fields** dialog box, select the fields from the case type data model that you want to import to process in the consumer system.
  - c. Click **Submit**.
8. Click **Save**.

At run time, your consumer application can process cases of the selected case type from the producer application.

[Identifying case types elements \(on page 9\)](#)

[Building case types \(on page 20\)](#)

[Configuring a data model for a case \(on page 150\)](#)

## Updating connection details for remote case types

To ensure that the connection between your consumer and producer applications works correctly after you make changes in your producer system, update the connection details in the consumer system. For example, if you need to change the URL of your producer system, reflect that change in the connection configuration. You can change the system URL, client ID, and client secret.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Connection** tab, in the **Connection details** section, click **Edit connection**.
3. In the **Edit connection** dialog box, edit the details as required, and then click **Submit**.
4. Click **Save**.

Key management system for application data encryption (on page )

Cosmos React (on page )

Authentication (on page )



## Error messages for remote case types

If any issues connected with remote case types occur, analyzing error messages helps you understand the cause of the issue and then make informed decisions about how to fix the issue. The JSON response to an issue includes relevant information that you can use when troubleshooting remote case types.

### JSON response

The following JSON response is common to all issues connected to remote case types that might occur in your system:

```
{
  "errorClassification": "error classification",
  "localizedValue": "the root cause of the issue",
  "errorDetails": [
    {
      "message": "error message type",
      "erroneousInputOutputFieldInPage": "",
      "erroneousInputOutputIdentifier": "",
      "errorClassification": "",
      "localizedValue": "The error message that an application displays at run time.",
      "messageParameters": [
      ]
    }
  ]
}
```

The following JSON response includes the details of a sample execution error:

```
{
  "errorClassification": "Execution error",
  "localizedValue": "Exception while getting access token",
  "errorDetails": [
    {
      "message": "Error_Execution_Error",
      "erroneousInputOutputFieldInPage": "",
      "erroneousInputOutputIdentifier": "",
      "errorClassification": "",
      "localizedValue": "Error while executing request. Please contact administrator.",
      "messageParameters": [
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

## Logs in fixing issues

Ensure that you enable logs so that you can obtain additional details about issues that might occur in your system. For remote case types, ensure that you enable `RemoteWebGatewayImpl` and `com.pegasys.platform.web.gateway.validations`. For more information, see [Log files tool](#) (*on page* ).

## Incoming requests errors

The following table lists the errors that are related to processing incoming requests in the consumer system:

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
Invalid tenant hash in the input URL	Tenant hash in the URL of the producer system is invalid.	Check the tenant hash in the URL of the producer system that you provide when you create a remote case type or update remote case type details.
Invalid request	The AAT cookie or the JWT token is invalid.	Ensure that required credentials are present and correct in the request.
Expired JWT Token	The JWT token that you use is expired.	Enable logs and contact Pega Support.
Empty remote system ID in the request	The <code>remotesystemid</code> header is empty in the request.	Verify that the <code>remotesystemid</code> header is present in the request.
Remote system details do not exist	The <code>data-admin-remotesystem</code> data instance is empty in the <code>remotesystemid</code> element of the request.	Verify that the <code>data-admin-remotesystem</code> data instance that corresponds to the <code>remotesystemid</code> element ex-



The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
		ists in your system and is correct.
AAT cookie not found in the request	The AAT cookie is missing from the request.	Verify that the AAT cookie is part of the request.
Invalid AAT	The exception while processing the incoming AAT token occurred.	Enable logs, contact your security team, and contact Pega Support.
Invalid URI request	URI in the remote system details is invalid.	Verify that the URI of the remote system that you provide when you configure the consumer system is correct.
Invalid request method	The REST calls do not support the requested HTTP method.	Ensure that you invoke only the supported HTTP methods.
Invalid CSRF Token	The CSRF token is invalid.	Enable logs, contact your security team, and contact Pega Support.
AAT Cookie is not parsed successfully	The system is unable to parse the incoming AAT cookie.	Enable logs, contact your security team, and contact Pega Support.
SSL Handshake exception	TLS version in authentication profile is not compatible with the system.	Change <b>Lowest allowable SSL/TLS version</b> to <b>TLS version 1.2</b> in the remote system authentication profile.

## Gateway processing errors

The following table lists errors that are related to gateway processing exceptions:

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
Exception when write stream from remote response to servlet response	Exception occurred while streaming a response from the remote system.	Enable logs and contact Pega Support.

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
Exception while firing the remote request	Not applicable.	Enable logs and contact Pega Support.
Connection pool shutdown while firing the remote request	The dynamic system setting for connection pool changed during request processing, and the request processing failed.	After you save the dynamic system setting with the new value, restart the request that failed.
Exception while firing request to remote system	Not applicable.	Enable logs and contact Pega Support.
Exception while getting connection from connection pool	The system failed to get the connection to fire the request.	Check logs for connection pools statistics and increase the connection pool size if necessary.
Could not get token from remote system	The token that the security settings return is empty.	Enable logs and contact your security team. Ensure that endpoints in the remote system run correctly.
Exception while getting access token	Security module returns an exception.	Enable logs, contact your security team and Pega Support.
Exception while renewing access token	Security module returns an exception on renewing access token.	Ensure that the client ID and client secret that you use to create remote case types are correct. Ensure that the endpoint in the remote system is reachable. If the client ID, client secret, and the endpoint work correctly, contact Pega Support.
Error while building socket factory for request system	The <code>remotesystemid</code> element has an invalid configuration.	Ensure that the configurations for tenant, keystore, and truststore are correct. If the configurations are correct, contact Pega Support.
Authorization has failed at remote system	The remote or producer system gave a 401 response.	Ensure that the client ID and client secret that you use to create remote case types are cor-

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
		rect. Ensure that the endpoint in the remote system is reachable. If the client ID, client secret, and the endpoint work correctly, contact Pega Support.
Exception while reading post data from request	An error with post data from the request occurred.	Verify the post that the request contains. If the data is correct, contact Pega Support.
Exception while parsing request content type	The request includes unsupported character encoding, or an exception occurred when parsing post data in the incoming request.	Verify the incoming request for unsupported character encoding. If the request does not include unsupported character encoding, contact Pega Support.

Configuring dynamic system settings (*on page* )

Troubleshooting tools and techniques (*on page* )

Authentication (*on page* )

Understanding authorized access tokens (*on page* )

## Configuring and tailoring case types

Provide users of your application with the ability to manage their work by creating case types that best suit their needs. Pega Platform offers a wide range of tools to modify case types and customize complex work processes to help users reach their business goals.

By changing a case status in the life cycle, you can communicate with stakeholders to ensure them that the process moves within the expected time frame.

### Configuring case type settings

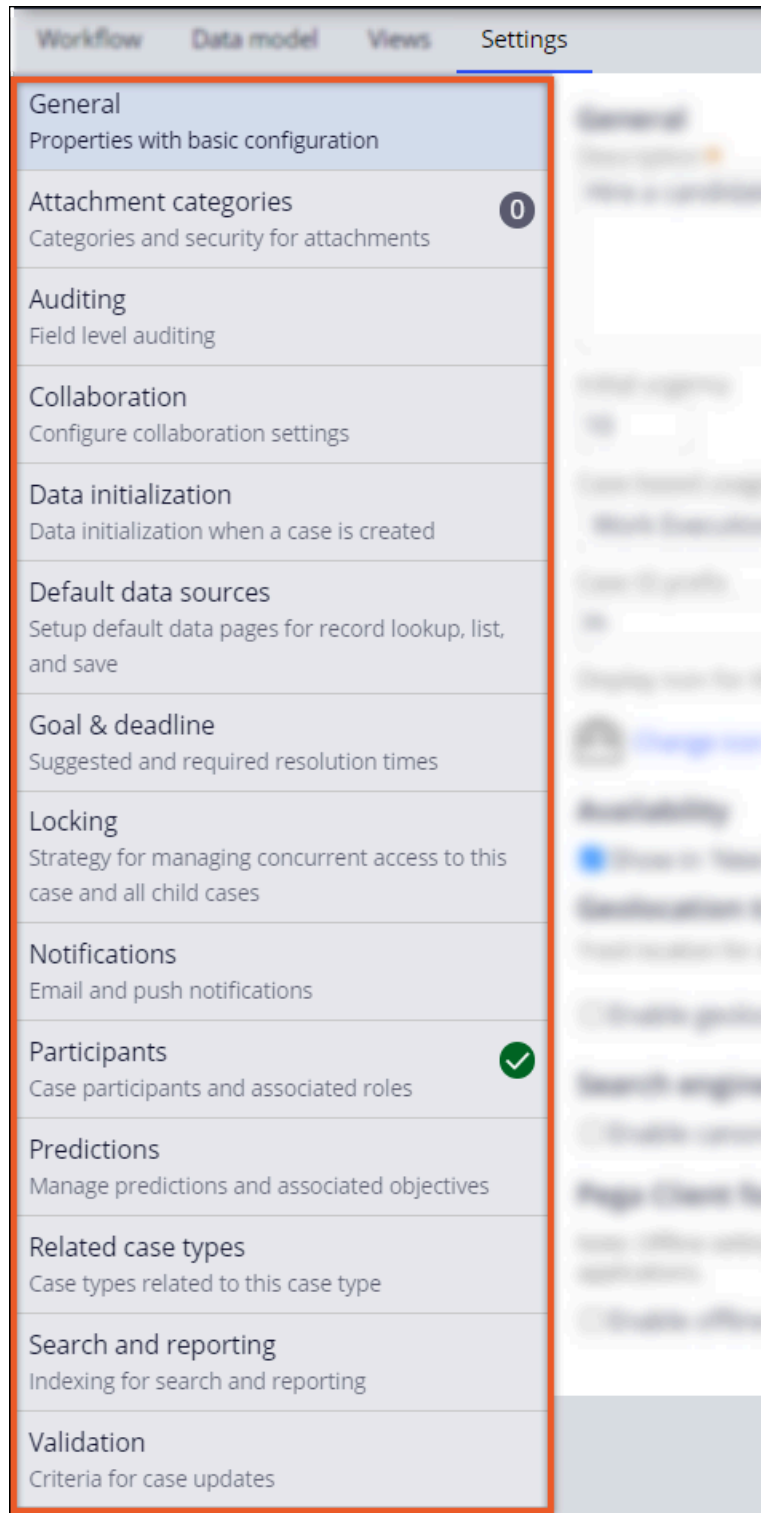
Better address the individual needs of your customers by configuring case types that precisely reflect their business processes. To provide the most efficient solutions for your customers, configure their case type settings to meet all of their requirements as closely as possible.

When you properly configure case type settings, you ensure that users of your application have all the tools that they need to successfully resolve their work. For example, you can define various case participants and their roles in a Microjourney®. By adding categorized attachments, participants can provide the



meaningful data and information necessary to resolve a case. To save time and improve efficiency, you can also enable automatic case creation and sharing of data between cases.

The following figure shows settings categories that you can configure for a case type:



## Configuring general case type settings

Configure general case type settings, such as customizing a case icon or changing a case ID format, to adjust cases to your unique business needs.

### Selecting a license category for a case type

Communicate the applicable business use case for your case type by selecting a relevant license category. The system uses license categories to report usage of your case types, and to help you ensure that you comply with your software requirements. For example, you can select a license category that indicates that most of the case processing occurs in an internal system.

1. In the navigation pane of App Studio, click **Case types**.
2. In the case work area, click the **Settings** tab.
3. In the navigation pane, click **General**.
4. In the **License category** list, assign a license to your case type:
  - To communicate that an application routes or tracks work in an internal system, select **Work Execution**.
  - To communicate that an application routes, tracks, and completes work by updating information in external systems, select **Work Management**.
  - To communicate that work processing occurs outside of an application, select **Work Dispatch**.
  - To communicate that work execution requires a custom extension for complex scenarios, select **Complex Work Execution**.
5. Click **Save**.

The case type has a license category but the default audit reports do not evaluate the classification. For more information about how these classifications affect your license agreement, contact your Pega Account Executive.

Licenses (on page )

License compliance (on page )

Classifying rules for license compliance (on page )

## Customizing the case icon

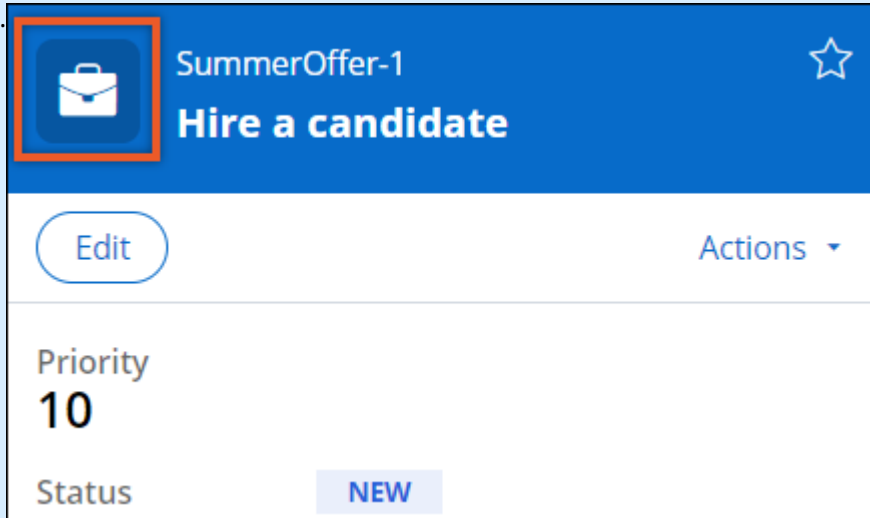
Customize the icon for each case type in your application to help users identify cases more quickly.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.



3. On the **Settings** tab, click **General**.
4. Click **Change icon**.
5. Click the icon that you want to select for your case type, and then click **Done**.
6. Click **Save**.

The icon that you select overrides the previous icon provided on the Application Form.



The screenshot shows a case card with a blue header. The header contains a briefcase icon (highlighted with a red box), the text 'SummerOffer-1', and a star icon. Below the header, the main title is 'Hire a candidate'. The card has an 'Edit' button on the left and an 'Actions' dropdown on the right. Below this, the 'Priority' is '10' and the 'Status' is 'NEW'.

Case icon at run time

#unique\_270 (on page )

[Changing the case ID format \(on page 206\)](#)

## Changing the case ID format

Change the case ID format to help case workers distinguish between instances of different case types. Custom case IDs can help users save time and simplify the organization of work by improving readability. For example, case IDs of the type Loan Request start with the prefix `LORE-`, and case IDs of the type New Account start with the prefix `NACC-`.

For each case, your application generates a unique case ID in the following format: *prefix-integer*. By default, a case ID contains the following elements:

### Prefix

A single character that your application derives from the name of the case type that you provide when you create or import a case type, and a hyphen. Prefixes are required, but they do not need to be unique.

More than one case type can share a prefix.

### Integer



A number that increases by one each time that you create a case. You cannot count the number of cases in your application based on the integer of a case ID because the system reserves case IDs before successfully creating a case.

The following examples of case IDs illustrate the different formats that you can use:

- C-100
- JobApp-420
- MORT-763

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **General**.
3. In the **Case ID prefix** field, enter a prefix for your case ID, for example, JobApp-.

When you enter the prefix, consider the following guidelines:

- A case ID prefix can consist of a maximum of 32 characters, including a hyphen at the end. If you do not enter a hyphen at the end of the prefix, the system adds it automatically after you save the settings.
- Do not use the ampersand (&) and double quotation marks (") in the case ID.

4. Click **Save**.

The following figure shows a new case with an ID that starts with the prefix that you configure for the case

type:

Initial urgency  
10

Case-based usage license tracking category  
Work Management

Case ID prefix  
SummerOffer- This is application specific.

Display icon for the case type  
Change icon

SummerOffer-1  
Hire a candidate

Edit Actions

Priority  
10

Status **NEW**

Setting a case ID prefix

#unique\_143 (on page )

[Customizing the case icon \(on page 205\)](#)

## Tracking case workers based on geolocation data

Record the location of users who work on specific assignments by enabling location tracking for case types. For example, enable location tracking in tasks for service employees in the field so that dispatchers can more effectively assign work to the employees based on their current location.



**Note:** Only standard Pega Platform applications support geolocation tracking. Applications that you build on Cosmos React do not support this feature.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. Click the **Settings** tab.



4. Click **General**.
5. Select the **Enable geolocation tracking** check box.
6. Click **Save**.

---

[Enabling field-level auditing \(on page 212\)](#)

## Disabling geolocation tracking

Disable geolocation tracking in case types in which it is not necessary to track the location of a case worker in the audit trail of a case. For example, you can turn off tracking for all Auto Loan cases because they are processed in the same office.

The standard *pyGeolocationTrackingsEnabled* when condition is specialized by class the first time that you enable geolocation tracking for a case type. As a result, all audit trail entries include a geographic location. You can disable this feature in the settings for a case type or provide a different set of conditions to evaluate.

The system applies each override to an individual case type or all case types, depending on the position of the *pyGeolocationTrackingsEnabled* when condition in the class hierarchy.



**Note:** You can disable geolocation tracking for all case types at the same time by deleting all instances of the *pyGeolocationTrackingsEnabled* when condition in your application. At run time, the system evaluates the *@baseclass.pyGeolocationTrackingsEnabled* version and then always returns a false value.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. Click the **Settings** tab.
4. Click **General**.
5. Clear the **Enable geolocation tracking** checkbox.
6. Click **Save**.
7. Click **Done**.

## Related information

[Enabling field-level auditing \(on page 212\)](#)

[Tracking case workers based on geolocation data \(on page 208\)](#)

[Specializing a case type \(on page \)](#)

## Creating temporary cases

To resolve unique cases that you do not intend to persist, and to save database resources, create a temporary case. For example, you can create a temporary case that tracks the delivery status of a package. This case is useful when customers want to know when their package arrives, but after the customer receives the package, you do not need to persist the case.

Your application cannot route a temporary case, and only one operator can process and resolve the case.



**Note:** You can only create a temporary case for case types without the default Create stage. Case types that you create in Pega Platform 8.5 and later have the Create stage included by default. You can only create a temporary case for case types that you migrate from Pega Platform 8.4 or earlier. For more information, see [The Create stage \(on page 48\)](#).

For relevant training materials, see the [Creating and persisting temporary cases](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click the **Settings** tab.
4. In the navigation pane, click **General**.
5. In the **Behavior** section, select the **Create temporary case that is not saved until a 'Persist case' step is reached** check box.
6. Click **Save**.

[Searching duplicate cases \(on page 109\)](#)

## Creating temporary cases on a case type rule form

To complete your very specific business scenario, you can create temporary cases on a case type rule form in Dev Studio for case types that include the Create stage. Temporary cases help you save database resources because the system does not store any data information about such cases until they become



persistent. For example, you can create a temporary case that at run time starts when a customer opens a portal to search for a flight to book. Then you can configure the system to persist the case only after the customer clicks a button to book a flight.

**Note:** All case types that include the Create stage are persistent by default. Temporary cases do not support various capabilities that are available for regular case types, such as attachments, service-level agreements, child case types, and routing assignments to specific users. As a best practice, avoid creating temporary cases. If you do not want to store cases that customers abandon at an early stage, configure a cleanup and archival policy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. Click the **Advanced** tab.
3. In the **Temporary case (Deprecated)** section, select the **Create temporary case that is not saved until a 'Persist case' step is reached** check box.
4. Click **Save**.

The system saves the case in the database after the case reaches the Persists case shape.

[The Create stage \(on page 48\)](#)

[Attaching content to a case \(on page 101\)](#)

[Understanding case hierarchy \(on page 14\)](#)

[Creating temporary cases \(on page 210\)](#)

[Completing work on time \(on page 324\)](#)

[Managing work across your team \(on page \)](#)

## Configuring collaboration settings for case types

Complete your case types faster by enabling collaboration tools such as Pulse messaging, case tagging and following.

Manage configurations for collaboration by performing the following actions:

### Enabling case tags

By enabling users to tag cases at run time, you speed up case search. When users use tags to filter cases, they retrieve relevant information faster.

Users can filter cases in several ways, for example, by displaying only the recently added or most frequently used tags. To find specific cases, users can also apply precise tags instead of choosing default filters. For more information, see [Filtering tagged cases \(on page 487\)](#).

**Note:** For applications that you build with the traditional Theme UI-Kit, see [Enabling case tags](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. Click the **Settings** tab, and then, in the navigation pane, click **Collaboration**.
4. In the **Tags** section, select the **Enable tags** checkbox.

At run time, users can add tags directly in the case view.

## Enabling field-level auditing

Maintain case compliance and follow progress in your case types by enabling field-level auditing. When you enable fields for auditing, users of your application can conveniently track how values have changed, such as an estimated end date, so that they can plan and process work more efficiently.

When you enable field-level auditing, your application tracks changes that users make to fields in the data model that your case contains. Users of your application can then check previous and current field values, the last user to change the values, and the time of their modification.

For relevant training materials, see the [Tracking and auditing changes to data](#) module and [Auditing changes to the value of fields](#) challenge on Pega Academy.

**Note:** Only standard Pega Platform applications support field-level auditing. Applications that you build on Cosmos React do not support this feature.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click the **Settings** tab.
4. In the settings pane, click **Auditing**.



5. In the **Auditing** window, turn on the **Enable field audit** switch.
6. In the **Fields** column, select the checkboxes next to the fields that you want to enable for auditing.



**Note:** Field-level auditing does not support data reference, page group, value group, and value list field types.



**Note:** If an application tracks changes in an embedded data field and the scope of the changes is greater than 66%, the application recognizes the changes as delete and add actions, and displays relevant audit information.

7. If you want to enable fields in an embedded data field, click the name of the field, and then repeat step 6 (on page 213).
8. Click **Save**.

At run time, when users process the case, they see the changes to the fields on the **Audit** tab.

[Configuring a data model for a case \(on page 150\)](#)

## Packaging rules and data that contain field-level auditing information

To save time and maintain consistency when you deploy your application in another environment, ensure that the field-level auditing data in your cases persists, by packaging specific rules and classes.

For example, when you move your application from a staging to a production environment, you can move the auditing data for the fields in your case types at the same time.



**Note:** To ensure optimal migration of data when you package an application, exclude non-versioned rules from the content of product rules, and include the rules that carry field-level auditing data as individual instances. For more information about packaging your application rules, see [Specifying the content of a product rule \(on page 150\)](#).

Ensure that you package the following rules of a *Rule-Obj-Class* rule type:

- A full class of a case type with enabled field-level auditing, for example *FLAudit-PegaProjMgmt-Work-UserStory-Epic*
- A class group of a class with enabled field-level auditing, for example *FLAudit-PegaProjMgmt-Work*



Ensure that you package the following data instances:

- DataAdmin-DataBase-ClassGroup, for example *FLAudit-PegaProjMgmt-Work*
- DataAdmin-DataBase, for example *FLAudit-PegaProjMgmt-Work*

---

Packaging your application in a product rule ([on page](#) )

## The extension point for field-level auditing

You can override the extension point for field-level auditing to facilitate reporting. For example, you can include the index keys of the audited properties in the audit table.

The extension point that you can use for field-level auditing in Pega Platform version 8.4 and later is the *pySave* activity from the *FLAudit-* class.

To learn how you can benefit from modifying the extension point for field-level auditing, see the following article with a sample use case:

---

### Related information

[Tracking and auditing changes to data](#)

[Auditing changes to the value of fields](#)

## Displaying the index keys of properties in the field-level auditing layout

Increase the readability of the field-level auditing layout by adding a column that displays the index keys of your audited properties of the Page List and Page Group types. When you expose the index column inside the section that is displayed for a property of the Page List and Page Group type, you can differentiate between the list items more conveniently.

1. Create and configure a property to represent the key of audited properties:
  - a. Create a property with the *IndexKey* name and of the Text date type in the *FLAudit-* class in any of the application rulesets.  
For more information, see [Properties](#) ([on page](#) ), [More about Properties](#) ([on page](#) ), and [Configuring page, page group, and page list properties](#) ([on page](#) ).
  - b. On the new property form, in the **Additional configuration options** section, select the **Optimize for reporting** and **Optimize for reporting on descendant classes** checkboxes.

For more information, see [Optimize a property from the user interface](#) ([on page](#) ) and the [Optimizing properties for report performance](#) challenge on Pega Academy.



The *IndexKey* property is now configured to support a new column for presenting key values:

The screenshot shows the configuration interface for the *IndexKey* property. The breadcrumb path is *CL: FLAudit- > ID: IndexKey > RS: EFarm [Branch: TestBugLog]*. The configuration is divided into several sections:

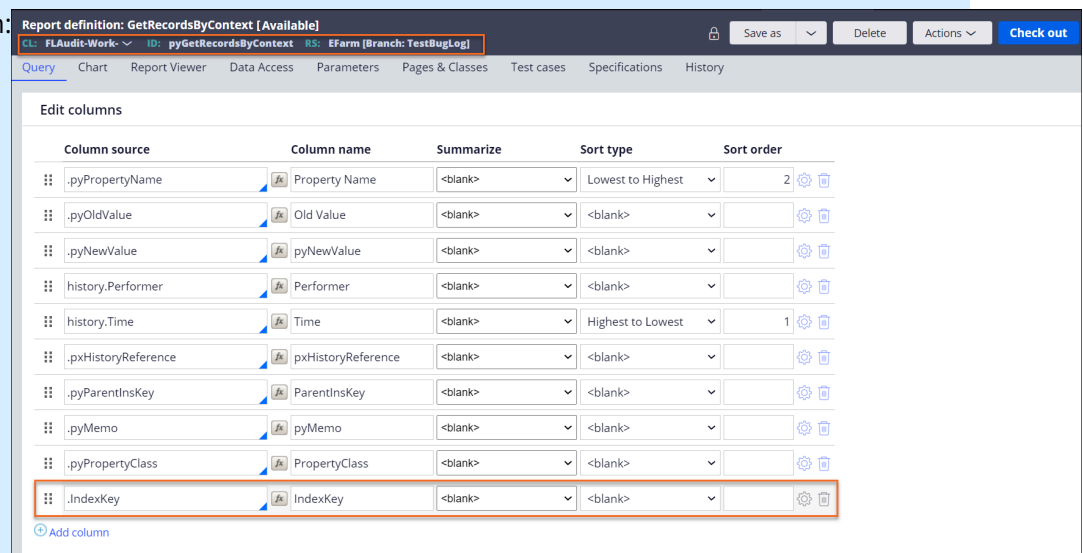
- Property type:** Text
- Data access:** Manual (selected), Automatic reference to class instance (linked) (unselected). A note states: "At run time, the user adds data to this property through the UI. Data transforms and other rules may be required to support this workflow."
- Display and validation:**
  - UI Control:** pxTextInput
  - Table type:** None

The *IndexKey* property



2. Override an extension activity to set the *IndexKey* property:
  - a. Expand the **Technical** category, and then click **Activity**.
  - b. Find and open the *pySave* activity from the *FLAudit-* class.
  - c. On the activity form, click **Save As**.
  - d. On the new activity form, in the **Apply to (class)** field, select the class that you created for the case type in the *FLAudit-* class.
  - e. In the **Add to ruleset** field, select the application ruleset that contains the selected class.
  - f. Click **Create and open**.
  - g. On the **Steps** tab, in the **Method** field, select a method, and then click **Expand to see method parameters**.  
The activity references *Param.ContextPage* as the work page and the FLA record as the step page.
  - h. Click **Save**.
3. Override a report definition that derives records from the *FLAudit* table:
  - a. Expand the **Reports** category, and then click **Report Definition**.
  - b. Find and open the *pyGetRecordsByContext* report definition from the *FLAudit-* class.
  - c. On the report definition form, click **Save As**.

- d. On the new report definition form, in the **Apply to (class)** field, select the class that you created for the case type in the *FLAudit-* class.
- e. In the **Add to ruleset** field, select the application ruleset that contains the selected class.
- f. Click **Create and open**.
- g. On the **Query** tab, click **Add column**, and then in the **Column source** field, enter `.IndexKey`.
- h. Click **Save**.

The following figure shows the `.IndexKey` property added to the overridden report definition:



The overridden `pyGetRecordsByContext` report definition

4. Include the new property in the table section that displays the FLA records:
  - a. Expand the **User Interface** category, and then click **Section**.
  - b. Find and open the `pyFieldGroupHistory` section from the `@baseclass` class.
  - c. On the section form, click **Save As**.
  - d. On the new section form, in the **Apply to (class)** field, select the class that you created for the case type in the *FLAudit-* class.
  - e. In the **Add to ruleset** field, select the application ruleset that contains the selected class.
  - f. Click **Create and open**.
  - g. On the **Design** tab, in the table area, click a table column, and then click **Insert column after selected** or **Insert column before selected**  .



- h. Next to the new column's label area, click **View properties**, and then, in the **Cell properties** pane, in the **Value** field, provide the name for the column. For example, enter `key`.
- i. Click **Submit**.
- j. Next to the new column's field area, click **View properties**, and then, in the **Cell properties** pane, in the **Property** field, enter `.IndexKey`.
- k. Click **Submit**.

The screenshot shows the 'Cell Properties' dialog box with the following details:

- Control inherited from property:** [Change](#)
- General** | Presentation | Actions
- Property:** `.IndexKey` (highlighted with an orange box)
- Default value:** (empty field)
- Visibility:** Always (dropdown menu)
- Disable:** Never (dropdown menu)
- Show when active
- Required

The new property can be exposed and captured as part of field-level auditing. In the presented example, the screen captures the `IndexKey` value and displays the value in the field-level auditing table.

## Sharing data between parent and child cases

For greater automation of your business processes, you can share data among the cases in your case hierarchy. When you reuse data between parent and child cases, you increase efficiency, improve data consistency, and as a result, resolve cases faster.

## Aggregating data in a parent case in Dev Studio

Speed up case processing by configuring a parent case to update specific values each time the values in child cases change. For example, you can create a parent case that represents a request for an insurance offer, and then supplement the case with child cases that visualize life insurance and property insurance

in an insurance application. When a customer service representative (CSR) updates the values in the child cases, the insurance application automatically calculates the total value in the parent case.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. Click the **Settings** tab, and then click **Calculations**.
3. In the **Calculations** section, click **Add calculation**.
4. In the **Calculated property** field, press the Down arrow key, and then select the property in which you want to aggregate data.

The calculated property stores the sum of other properties, across all child cases of the parent case.



**CAUTION:** Ensure that this property is not associated with a declare expression, because the value of the property can be overwritten.

5. In the **Case type** list, select a class of a case type that is a child of your current case type.
6. In the **Property** field, press the Down Arrow key, and select the name of a single-value property that you want to add to the cumulative property.
7. **Optional:** To add more properties that the system sums up at run time, click **Add property**, and then repeat steps [5 \(on page 218\)](#) through [6 \(on page 218\)](#).
8. Click **OK**.
9. Click **Save**.

At run time, an application updates the value of the calculated property in the parent case each time you change any of the selected properties in a child case.

[Sharing data between parent and child cases \(on page 217\)](#)

Declare Trigger rules (on page )

## Propagating data to a child case

Resolve cases faster by propagating data from a parent case every time you create a new child case. By sharing data between cases, you save time and ensure that your application uses consistent data throughout the entire case life cycle. For example, you can create a parent case type that represents an accident claim review, and then supplement the process with child case types that visualize the estimation of vehicle damage and injuries. When you then create child case types, your application automatically copies the personal details of the customer issuing the claim into the new child case types. As a result, you save time because the information is already in the child cases.



To propagate data, you first define the source properties in a parent case type that store the information that you want to copy, and then define the target properties in the child case type in which you want to reuse the data.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. Click the **Settings** tab, and then click **Data propagation**.
3. In the **Data propagation** section, click a child case in which you want to reuse data.
4. Click **Add property**.
5. In the **Propagate property value** field, press the Down arrow key, and then select the source property from which to copy data.
6. In the **To property value** field, press the Down arrow key, and then select the name of the property that your application sets in the child case at run time.
7. Click **OK**.
8. **Optional:** To propagate the values of more properties from the parent case type, repeat steps 4 (on page 219) through 7 (on page 219).
9. **Optional:** To apply a data transform to set property values:
  - a. Select the **Apply data transform** check box, and then select a data transform to apply.
  - b. On the **Definition** tab of the data transform, in the **Source** field, enter `pyParentPageName`.



**Note:** Do not add `param` before the page property name because the parameter is of the Page name type.

- c. On the **Pages & Classes** tab, in the **Page Name** field, enter `pyParentPageName`.
- d. In the **Class** field, enter a class that is equal to the class of the parent case.
- e. On the **Parameters** tab, add a new entry with a name of `pyParentPageName`.
- f. In the **Data type** field, select **Page name**.



**Note:** Do not mark the parameter as required because the system provides the value at run time.

10. Click **Save**.

At run time, when you create a child case type, your application propagates the values of the properties that you select.

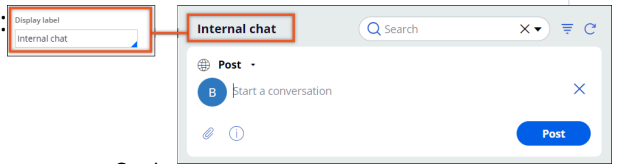
[Sharing data between parent and child cases \(on page 217\)](#)

[Creating contextual cases \(on page 106\)](#)

## Configuring Pulse for case types

Provide case workers, such as customer service representatives (CSRs), with a collaboration tool for open discussion of their work, by configuring Pulse. To accelerate case type development, configure Pulse directly in the case type settings. When you enable Pulse, case workers can post messages to their colleagues and receive replies. To ensure that case workers only see relevant messages, you can define default feed sources and post display conditions.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click the **Settings** tab.
4. In the **Pulse** section, configure the Pulse settings:

Choices	Actions
<b>Change the default label</b>	<p>In the <b>Display label</b> field, enter a new label.</p> <p>For example, enter <b>Internal chat</b>, as in the following figure:</p>  <p>Setting the Pulse gadget label for a case</p>
<b>Enable posting Pulse messages</b>	Select the <b>Allow users to add posts</b> check box.
<b>Define a condition for post visibility</b>	In the row of a post type that you want to configure, in the <b>Visibility</b> list, select when the application displays posts.
<b>Add a feed source</b>	<ol style="list-style-type: none"> <li>a. In the <b>Configure additional feed sources</b> section, click <b>Add Pulse feed</b>.</li> <li>b. In the <b>Pulse feed name</b> list, select the source that you want to add.</li> </ol>

Choices	Actions
	<p>c. <b>Optional:</b> To display the feed source default, select the <b>Display feed by default</b>.</p> <p>d. Click <b>OK</b>.</p>

5. Click **Save**.

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

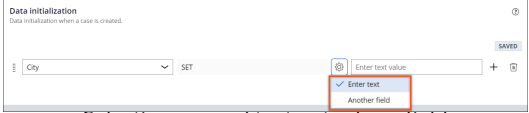
[Collaborating with users by using Pulse \(on page 499\)](#)

## Initializing data at case creation

Increase the efficiency of customer service and minimize the need for customer input by initializing data for new cases. As a result, your application preloads information that you might need at run time, such as location, and uses it to complete the fields when a case starts. For example, a case type for a mobile application that is available in only one city can use that city as the default location to provide a relevant offer. Loading data automatically for a new case saves time for citizen developers and for customers.

You initialize data for a case type in App Studio by authoring a data transform in the case type settings. A data transform passes data from one field to another. For each field that you want to prepopulate in the Create stage whenever you create a case, you either set the value manually or select another field as the source of the value. To configure more advanced data transforms, you can switch to Dev Studio directly from the Data initialization landing page.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **Data initialization**.
3. In the **Data initialization** section, in the **Select field** list, select the field for which you want to set the value at case creation.
4. Click the **Choose source type** icon, and then select the source of the value for the field:

Choices	Actions
<p><b>Manually enter the value for the field</b></p>	<p>a. Click <b>Enter text</b>, as shown in the following figure:</p>  <p>b. In the <b>Enter text value</b> field, enter the text that you want to use as the field value.</p>
<p><b>Use the value of another field</b></p>	<p>a. Click <b>Another field</b>.</p> <p>b. In the <b>Select field</b> list, select the field whose value you want to use.</p>

- Optional:** To initialize an additional field, click the **Add row** icon, and then repeat steps 3 (on page 221) and 4 (on page 221).
- Click **Save**.

### Related information

Configuring a data transform (on page )  
 Configuring case type settings (on page 203)

## Initializing data at case creation in Dev Studio

Automatically load data for a new case by configuring data initialization for a case type. For example, a mobile application uses a customer's location data to provide products that are available in the customer's area. As a result, customer service is more efficient because the customer does not need to manually provide additional information each time they use the app.

- In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
- On the **Settings** tab, click **Data initialization**.
- In the **Data initialization** section, click **Configure in Dev Studio**.
- Configure the data transform that you want to run at case creation.  
 For more information, see Data transforms (on page ).
- Click **Save**.

The system runs the *pyDefault* data transform to initialize data whenever you create a case, and uses this data to populate the fields in the Create stage.

### Related information

Configuring a data transform (*on page* )

Configuring case type settings (*on page 203*)

## Configuring case creation

You can control when cases are created by configuring one or more creation methods in a case type. By default, users can manually create top-level cases only.

Use the following techniques to configure the creation methods for each case type in your application:

Creating an Email channel (*on page* )

## Enabling automatic creation of child cases

Enable automatic creation of a child case type each time that a parent case is created.

Ensure that you open the correct version of your case type if it has multiple parent case types. The instantiation details for a child case type are unique to each parent case type.

**Note:** The Instantiation tab is available only for child case types in Dev Studio.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **Instantiation**.

**Note:** The **Instantiation** tab is available only for child case types in Dev Studio.

3. Select the **Automatically by system when** check box.
4. Click **Parent case starts**.

If your case type has more than one parent case type, the parent is determined by the navigation path that you followed in the Case Type Explorer to open the case type.

5. **Optional:** To refine the conditions that control whether a case is created, press the Down Arrow key in the **Allow only when** field, and then select the name of a when condition.
6. Click **Save**.

At run time, your application creates a child case when it detects that a parent case is created and the when condition that you provide returns a value of true.

[Enabling automatic conditional creation of child cases \(on page 224\)](#)

[Enabling creation by email of top-level cases \(on page 226\)](#)

[Configuring case creation \(on page 223\)](#)

## Enabling automatic conditional creation of child cases

Enable conditional creation of a child case after creating a parent case when the case data meets the conditions that you define. For example, to save time and automate work, you can configure an application that processes purchase orders to create a shipment case when the status of its parent case, which tracks the order and payment, changes to `Billing-Approved`.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **Instantiation**.

**Note:** The **Instantiation** tab is available only for child case types in Dev Studio.

3. Select the **Automatically by system when** check box.
4. In the **of the below conditions are met** list, select how many conditions the case data needs to meet to create a child case:
  - To create a child case when case data meets any of the conditions, select **Any**.
  - To create a child case when case data meets all of the conditions, select **All**.
5. Define a dependency by clicking **Add condition**.
6. In the **Case** field, press the Down arrow key, and then select a case type on which you want your current case type to depend.
 

This list displays case types that share your topmost case type and excludes specialized case types, descendant case types, and case types with more than one parent.
7. Select the criteria for creating a child case:



Choices	Actions
Create a child case when a parent case starts	<ol style="list-style-type: none"> <li>In the <b>Condition</b> list, select <b>has started</b>.</li> <li>Click <b>OK</b>.</li> </ol>
Create a child case when a parent case reaches the specified work status	<ol style="list-style-type: none"> <li>In the <b>Condition</b> list, select <b>has work status</b>.</li> <li>Select a case status in the list of available statuses.</li> </ol> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p><b>Tip:</b> When using work status as a dependency condition, ensure that your process has at least one shape, such as an assignment or utility, that sets the status of a case to the expected value.</p> </div> <ol style="list-style-type: none"> <li>Click <b>OK</b>.</li> </ol>
Create a child case when at least one parent case completes	<ol style="list-style-type: none"> <li>In the <b>Condition</b> list, select <b>has completed</b>.</li> <li>Click <b>any</b> or <b>all</b> to indicate how many cases need to resolve before your application creates a child case.</li> <li>Click <b>OK</b>.</li> </ol>

- Optional:** To enable case creation only when the specified when condition evaluates to true, in the **Allow only when** field, press the Down arrow key, and then select the name of a when condition.
- Click **Save**.

At run time, your application creates a new child case when case data meets your dependency conditions, and the when condition that you provide returns a true value. The limitation of one child case prevents duplication when case data fulfills a dependency more than once, such as a parent case type reaching a specific stage.

[Changing case statuses \(on page 318\)](#)

[Configuring case creation \(on page 223\)](#)

## Enabling creation by email of top-level cases

Associate an email account with a top-level case type to provide a way for users to create cases remotely.

1. Configuring outbound email in Dev Studio ([on page 223](#)).

You do not need to create an email service implementation, because your application generates the required rules when you associate the email account with a case type.

2. Verify that you can connect to the email server.
  - a. Open the Email Account form.
  - b. In the **Sender** section, click **Test Connectivity**.
  - c. In the **Receiver** section, click **Test Connectivity**.
3. Use Admin Studio to ensure that the listener that supports your email account is running. For more information, see [Managing listeners \(on page 223\)](#).
4. Manually associate the email account with a top-level case type.



**Tip:** To increase the visibility of this association, use a channel interface instead. See [Setting up the Email channel \(on page 223\)](#).

- a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
- b. On the **Settings** tab, click **Email instantiation**.



**Note:** The **Email instantiation** tab is available only for parent case types.

- c. In the **Starting process** list, select a flow that runs when the case is created.

Although case types can have more than one starting process, an email account is associated with only one starting process at a time.



**Note:** A starting process applies only to case types without the Create stage. For more information, see [The Create stage \(on page 48\)](#).

- d. In the **Email account** field, press the Down Arrow key and select the name of an email account that is not associated with another case type in your application.

**CAUTION:** You are warned when you select an email account that is already in use. If you save your case type with this configuration, more than one listener monitors the same email account, which can result in inconsistent case creation.

- e. Select the **Enable email listener** check box to start the listener that monitors your email account.
  - f. Click **Save** to create the required integration rules.
  - g. **Optional:** For more information about the integration rules that support email instantiation, from Dev Studio, click **Configure > Integration > Email > Email Listeners**.
5. Test your changes by sending an email message to the address that is specified in the email account.

You can also reply to the confirmation email that you receive after the case is created, to continue the discussion with a case worker or ask a question about the case.

When your application creates a case in response to an email message, the original message and corresponding replies are stored in Pulse.

### Related information

[Collaborating with users by using Pulse \(on page 499\)](#)

[Notifying participants about events \(on page 235\)](#)

[Sending automatic emails from cases \(on page 142\)](#)


## Email account limitations

The following limitations apply to the email accounts in your application. Ensure that you understand how these limitations affect your application before you enable email approval or email instantiation for a case type.

- You can have only one email account that manages case notifications, such as requests for approval, in your application. This email account is either the standard `Default` account or the account with your work pool name.

As a best practice, create one email account per work pool and then use the `Default` account for validation.

- You can use one email account to support email approval for instances of one case type or all case types in your application. If you want to use unique email accounts, you must create additional applications and email accounts for each case type.
- You can enable email instantiation for only one case type in an application. Therefore, if you want users to create cases for more than one case type, you must create an application and email account for each case type.

**Note:** To minimize the steps that it takes to configure an email account, use the Email Wizard  because it automatically creates the rules and data instances that you need to manage incoming email messages.

---


[Collaborating with users by using Pulse \(on page 499\)](#)

[Configuring outbound email in Dev Studio \(on page \)](#)

## Creating a case by using an API

Increase the versatility of your Pega Platform™ application by creating a case by calling the create case microservice from an external application. Consequently, you improve the efficiency of your business processes and save time by automating case creation. For example, when a customer submits a ticket in a technical support application, the application creates a bug in your project management system, such as JIRA.

The **Generate create case microservice code** option opens a dialog box that contains the information that you can use in a Pega API call to call the create case microservice from your external application. By calling the microservice, you create a case in your application.

 **Tip:** For more examples of API-based communication that is typical of managing a case life cycle across environments, see [Calling sequences for DX API endpoints](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **Integration**.
3. In the **Microservice APIs** section, click **Generate create case microservice code**.
4. Create a case of the selected case type from your external application by using the information in the following fields:
  - a. In the **Request URL (HTTP POST method)** field, copy the URL that you can access to create the case, and then paste the URL in the external application. Ensure that you invoke the URL by using the HTTP POST method.
  - b. In the **HTTP authentication header** field, copy the header for the microservice call, and then, in the external application, enter the login credentials in the header for basic HTTP authentication.
  - c. In the **Request body** field, copy the body for the microservice call, and then, in the external application, paste the body and replace the values for fields in the content node with values that are appropriate for the case that you want to create.

For more information, see the documentation for your external application.

5. **Optional:** To see a list of the built-in Pega API REST services in Pega Platform, click **View all Pega API**.

---

[Creating cases from a web mashup \(on page 384\)](#)

[Enabling case processing in an external application \(on page \)](#)

## Defining default data sources

Provide explicit sources of information for your cases by defining default data pages that an application uses to retrieve case information. When you expand your application by creating objects that interact with your case type or data object, such as case references, those objects load data from the data pages that you set as default. As a result, data management in your application is more transparent and easier to analyze. For example, you can define `Contact information` as a default data page for an `Expand account` case type. Then, when you create objects that interact with this case type, the system prompts the `Contact information` data page as a default data source for the new object.



1. Navigate to the element of your application for which you want to define a default data source:

Choices	Actions
<b>Add default data sources to a case type</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of Dev Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. In the case working area, click the <b>Settings</b> tab.</li> <li>c. In the settings pane, click <b>Default data sources</b>.</li> </ol>
<b>Add default data sources to a data object</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Data</b>.</li> <li>b. In the <b>Data objects</b> section, click the data object for which you want to define default data sources.</li> <li>c. Click the <b>Settings</b> tab, and then in the settings pane, click <b>Default data sources</b>.</li> </ol>

2. Define the sources that your application uses to fetch information:

- To define a default data page that loads information, in the **Default record lookup Data Page** list, select a data page that you want to use.

You can select only data pages of a page structure.

- To define a default data page that loads information in the form of a list, such as a list of office locations that are available for users, in the **Default list Data Page** list, select a data page that you want to use.

You can select only data pages of a list structure.

- To define a savable data page that loads information and that you can use to update and save case information, in the **Default save Data Page** list, select a data page that you want to use.

You can select only savable data pages.

3. Click **Save**.

At run time, an application sources information from a default data page that you provide. Additionally, when you create a case reference, an application uses a default data page of a target case as a source of information. For more information, see [Referencing a case type \(on page 153\)](#).

[Data pages overview \(on page \)](#)

[Savable data pages \(on page \)](#)

[Configuring a data model for a case \(on page 150\)](#)

[Saving data in a data page as part of a case life cycle \(on page 148\)](#)

[Preloading a data page \(on page 136\)](#)



## Managing concurrent access to a case

Improve efficiency of your business processes and ensure that users can bring a case to a successful resolution by setting a locking strategy for parent case types to reduce the risk of losing updates to cases that users make simultaneously. For example, in an insurance claim review case, you can lock a case to ensure that the case can move to the next stage only after a customer service representative (CSR) provides all data necessary for a case resolution.



**Note:** Only parent case types support this option. However, all child cases inherit locking strategy from a parent case.

In most configurations, use the default locking strategy to preserve transaction integrity among cases. Without default locking, you can lose work when other users perform tasks such as bulk processing or escalation actions in a service-level agreement.

You can provide one user or multiple users with concurrent case access. Ensure that you adjust the locking strategy to your business scenario by considering the following factors:

- To maintain transaction integrity in both parent and child cases, enable one user to make data updates. For example, the parent case can contain properties that calculate or sum up values in the child case. Locking both parent and child cases at the same time helps to keep the counts or totals in sync.
- If multiple users need to open and review cases simultaneously without updating the cases, allow multiple users to access a case. Allow multiple users to update a case also in a scenario when a user needs to update a parent case shortly after updating a child case. In the background, the system locks a parent case for a short period of time after updating a child case, and if only one user can update a case, the system returns an error when the user updates the parent case.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Settings** tab, click **Locking**.
4. In the **Access strategy to open and work on a case** section, select one of the available locking strategies:

Choices	Actions
<p><b>Only one user can access a case</b></p>	<p>a. Select <b>Allow one user</b>.</p> <p>b. <b>Optional:</b> To adjust the time-out to your needs, in the <b>Time out value</b> <i>time-out duration mins.</i> field, enter how many minutes a lock remains on a case. The default value is 30.</p> <p>Your application locks the case for a time-out duration or until the user submits or closes the case.</p>
<p><b>Multiple users can access a case simultaneously</b></p>	<p>a. Select <b>Allow multiple users</b>.</p> <p>Providing multiple users with access to a case is necessary if you want users to access cases offline.</p> <p>Your application preserves the changes that the first concurrent user makes. All other users who work on the case receive notifications and must review the changes before they can submit any updates.</p>

5. Click **Save**.

The locking strategy applies to new and existing cases.

[Securing case access \(on page 453\)](#)

## Overriding the locking strategy of a child case

To meet your unique business needs and customize how your application manages concurrent updates to cases, override a child case locking strategy. By default, when the child case is open, an application puts a lock on a parent case. When you override the locking strategy of a child case, users can update the parent case even if the child case is open. As a result, users can resolve work faster and more efficiently.



For example, you can create a parent case type that represents reviewing an insurance claim after a car accident, and a child case type that processes vehicle damage. Then you can override the locking strategy of the child case and allow users to work on a car accident review even if a child case to process vehicle damage is already open.

Additionally, you can define a lock duration for a child case that is shorter than a lock duration for a parent case. For example, if a lock remains on a parent case for 30 minutes, you can define a lock for a child case that lasts for 20 minutes, so that users can start processing a locked child case quicker.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Settings** tab, click **Locking**.
4. Customize the locking strategy:

Choices	Actions
<b>Allow users to access a parent case when another user works on a child case</b>	Select the <b>Allow other users to access parent case when the child case is opened</b> check box.
<b>Change the default timeout duration</b>	In the <b>Time out value</b> <i>time-out duration mins.</i> field, enter how many minutes a lock remains on a child case.  <div style="border: 1px solid #0070C0; background-color: #D9E1F2; padding: 5px;"> <p><b>Tip:</b> Enter a time-out value for the child case that is lower than the time-out value of the parent case.</p> </div>

5. Click **Save**.

[Case locking](#)

[Managing concurrent access to a case \(on page 231\)](#)

[Securing case access \(on page 453\)](#)

## Locking stand-alone cases

Ensure data consistency by defining a locking strategy that applies in scenarios when a case type is a stand-alone case, a child case, or a parent case. By configuring multiple locking settings at once, you ensure that your case type meets access and security requirements in every scenario. For example, in an insurance application, you can define a locking strategy for a Body injury case type to define how locking works when the case type is a stand-alone case or a parent for a Review medical records case type. At the same time, you can configure locking behavior when the Body injury is a child of a Car accident case type. As a result, you save time because you provide configurations for different use cases at once.

You lock stand-alone cases by using a case type rule form in Dev Studio.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Case Type**.
3. Open a case type that you want to edit.
4. On the **Advanced** tab, in the **Locking** section, define the locking strategy when this case type is a parent case or a stand-alone case:

Choices	Actions
<p><b>Only one user can access a case</b></p>	<p>a. Select <b>Allow one user</b>.</p> <p>b. <b>Optional:</b> To adjust the time-out value to your needs, in the <b>Time out value time-out duration mins.</b> field, enter how many minutes a lock remains on a case. The default value is 30.</p> <div style="background-color: #e1f5fe; padding: 10px; border-left: 2px solid #0070c0;"> <p>Your application locks the case for a time-out duration or until the user submits or closes the case.</p> </div>
<p><b>Multiple users can access a case simultaneously</b></p>	<p>a. Select <b>Allow multiple users</b>. Providing multiple users with access to a case is necessary if you want users to access cases offline.</p> <div style="background-color: #e1f5fe; padding: 10px; border-left: 2px solid #0070c0;"> <p>Your application preserves the changes that the first concurrent user makes. All other users who work on the case receive</p> </div>

Choices	Actions
	<div style="background-color: #e1f5fe; padding: 5px;">           notifications and must review the changes before they can submit any updates.         </div>

5. **Optional:** To leave a parent case unlocked when this case type is a child case, select the **Do not lock the parent case when the child case is being performed** check box.

After resolving a child case, the system automatically locks a parent case for a short period of time to update relevant properties. If you leave a parent case unlocked and another user performs edits in the parent case when the system attempts to automatically acquire the lock, the error occurs. To prevent errors, allow multiple users to access the parent case.

6. Click **Save**.

### Related information

[Understanding case hierarchy \(on page 14\)](#)

[Case types \(on page 28\)](#)

[Securing case access \(on page 453\)](#)

## Notifying participants about events

Escalate expired service-level agreements and highlight cases that require immediate action by sending email and push notifications to case participants. You can use push notifications for mobile devices, and set up email notifications for all devices that support email. As a result, you ensure that business processes reach resolution within the defined time frame. For example, a case worker can receive an email and a push notification after a goal for an assignment expires, or after an application routes a new assignment to the user.

All assignments in a case life cycle inherit the notification policy from the case type. The following scenarios do not support notifications:

- You set the notification policy after creating an assignment.
- A user pulls an assignment from a work queue.

Users can receive notifications after an application routes an assignment to their worklists, but not after pulling assignments from shared work queues.

- A user transfers ownership of an assignment to another user.
- An application routes the assignment to the worklist of the user who already processes the case.



**Note:**

When your application sends an email, the system updates the *pxSendDateTime* property on the *pxCorrSummary* work page that belongs to *pyWorkPage*. Such updates might cause issues with refreshing a case at run time. To prevent errors, the *pyEnableCorrSummary* when rule is set to false by default, and the system does not update the *pxSendDateTime* property. If your business scenario requires updating the *pxSendDateTime* property, in Dev Studio set the *pyEnableCorrSummary* when rule to true.

- Correspondence rules (on page )
- [Disabling assignment notifications \(on page 240\)](#)
- [Sending custom notifications about new assignments \(on page 239\)](#)
- [Sending confirmations of completed assignments \(on page 239\)](#)
- Configuring outbound email in Dev Studio (on page )
- When condition rules (on page )

## Sending email notifications from cases

To speed up resolution of your business processes, ensure that users receive information when new tasks await in their worklists, by sending email notifications. For example, you can compose a message that includes a link to a case, so that a user can immediately access the case and start work.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case work area, click the **Settings** tab.
4. In the settings panel, click **Notifications**.
5. In the **Email notifications** section, select the **Email user when assignment is routed to worklist** check box.
6. Define the content of your message:

Choice	Actions
<b>Compose a new email</b>	a. In the <b>Email</b> section, select <b>Custom</b> .

Choice	Actions
<b>Use an existing email</b>	<ul style="list-style-type: none"> <li>a. In the <b>Email</b> section, select <b>Use existing</b>.</li> <li>b. In the <b>Message</b> list, select a message that you want to use.</li> </ul>

7. In the **Subject** field, enter the email title by entering a string expression.

You can reference fieldproperty names in the title to make the heading more dynamic and meaningful.

8. If you create a custom message, in the **Message** section, click **Compose**, and then edit the message in the rich text editor:

Choices	Actions
<b>Compose your own message</b>	<ul style="list-style-type: none"> <li>a. Enter and style the text by using the rich text editor toolbar.</li> </ul>
<b>Populate your email message with text from a template</b>	<ul style="list-style-type: none"> <li>a. In the <b>Compose message</b> window, click <b>Choose template</b>.</li> <li>b. Click a template that you want to use.</li> <li>c. Click <b>Select</b>. After the template text appears in your email message, you can edit this content.</li> </ul>
<b>Include the value of a field from your case in your email</b>	<ul style="list-style-type: none"> <li>a. In the rich text editor, position the text cursor in the place where you want to add a property.</li> <li>b. On the toolbar, click the <b>Insert property</b> icon.</li> <li>c. Select a property that you want to use.</li> </ul> <div data-bbox="938 1640 1463 1875" style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p><b>Tip:</b> To prevent word clusters in your message, include a space before and after your property reference.</p> </div>

Choices	Actions
<p><b>Provide instant access to the case from your email</b></p>	<ol style="list-style-type: none"> <li>In the rich text editor, position the text cursor in the place where you want to add a link.</li> <li>On the toolbar, click the <b>Link</b> icon.</li> <li>Select the <b>Link to current case</b> check box.</li> <li>Enter the display text and title for the link.</li> <li>Click <b>OK</b>.</li> </ol>

9. Click **Save**.

A user receives an email when an application routes a task to their worklist.

[Assigning tasks to users \(on page 84\)](#)

[Assigning users automatically at run time with business logic \(on page 85\)](#)

## Sending push notifications about case events

To improve the resolution of cases, ensure that mobile users of your application receive information about meaningful events in business processes, by enabling push notifications. You can notify users about new assignments in their worklists, and then ensure that the users resolve their tasks on time by sending goal and deadline notifications.

- In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
- In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
- In the case working area, click the **Settings** tab.
- In the settings navigation panel, click **Notifications**.
- In the **Push notifications (only for mobile app)** section, enable notifications about elapsed goals and deadlines by selecting the **Enable push for goal and deadline notifications** check box.

When you set at least one escalation action to notify the assignee or the manager, and then the service-level agreement for an assignment expires, the system sends push notifications to users whose devices are logged on.

- Enable notifications about new assignments by selecting the **Notify user when assignment is routed to worklist** check box.
- Click **Save**.



[Assigning tasks to users \(on page 84\)](#)

[Assigning users automatically at run time with business logic \(on page 85\)](#)

[Completing work on time \(on page 324\)](#)

## Sending confirmations of completed assignments

Inform users about completion of assignments by sending confirmation notes. Case workers, such as customer service representatives, receive confirmation notes when they complete an assignment and an application routes the assignment to another worker, or when a case moves to the next step in the life cycle.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, in the process that you want to edit, click **Configure process**.
3. In the flow canvas, double-click the assignment for which you want to add a notification.
4. In the **Advanced** section, click **Assignment details**.
5. In the **Confirmation note** field, press the Down arrow key, and then select the name of a field value that contains a message.

As a best practice, create confirmation notes that reflect the assignment instructions.

6. Click **Submit**.
7. Click **Save**.

At run time, the *Work-.pyConfirmMessage* section displays the confirmation note when the user completes an assignment, or when the case moves to the next step in the life cycle, and an application routes that step to a different user or a work queue. If a user processes two assignments sequentially, the application sends only the confirmation note for the assignment that a user completes later.

[Assignment shapes in processes \(on page 270\)](#)

## Sending custom notifications about new assignments

To send custom correspondence to case participants about the creation of new assignments, override your notification policy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, in the process that you want to edit, click **Configure process**.
3. In the flow canvas, double-click the assignment for which you want to add a notification.



4. In the **Assignment properties** dialog box, in the **Advanced** section, expand the **Notifications** section.
5. In the **Send notification** list, select **Custom**.
6. In the **Notify** field, press the Down arrow key, and then select an activity that sends correspondence to one or more work parties.
7. If the activity accepts parameters, enter a value for each parameter in the fields that appear. Your application validates the parameter values when you save the flow.
8. Click **Submit**.
9. Click **Save**.

When a case reaches the assignment, an application sends a notification with contents and participants that you define in an activity that the notification uses.

[Notifying participants about events \(on page 235\)](#)

## Disabling assignment notifications

Prioritize the tasks that need the greatest visibility in a case by disabling notifications for other assignments.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, in the process that you want to edit, click **Configure process**.
3. In the flow canvas, double-click the assignment for which you want to add a notification.
4. In the **Assignment properties** dialog box, expand the **Notifications** section.
5. In the **Send notification** list, select **Never**.
6. Click **Submit**.
7. Click **Save**.

[Notifying participants about events \(on page 235\)](#)


## Configuring case participants

Identify the people, businesses, and organizations that receive case notifications by defining case participants. For example, you can create a group of customer service representatives so that they can receive notifications about new assignments and upcoming goals and deadlines. You can define internal participants that are users of your application, and external participants that do not have an account in your application, such as stakeholders. You can also refer to participants as work parties.





1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click the **Settings** tab.
4. In the settings pane, click **Participants**.

 **Note:** At run time, the widget is called Stakeholders.

5. In the **Participants** section, click **Add participant**.
6. In the **Participant configuration** window, in the **Role name** field, enter a unique name that indicates the relationship of the participant to the case type, for example, `worker`.
7. In the **Type** list, select the role that describes the participant:
  - To indicate that the participant is an application user, select **Operator**.
  - To indicate that the participant has a different role in the case, external to an application, select one of the options that describes the participant, for example **Person** or **Government**.
8. **Optional:** To provide information about the participant, such as a name or an email address, in the **Data transform** field, press the Down arrow key, and then select the name of a data transform that stores the participant details.
9. Define role preferences for the participant:

#### Choices

**Participant with an account in your application**

#### Actions

- a. In the **Role has user account with this application** section, select **Yes**.
- b. **Optional:** To populate information about a participant, in the **Preferences** section, associate the participant with the current user or a reporting manager of the current user, by clicking **Map participant**, and then selecting either **Current user** or **Reporting manager**.
- c. **Optional:** To save time and create more participants that have the same notification settings, enable creation of multiple participants for this role by se-

Choices	Actions
	<p>lecting the <b>Allow multiple participants for this role</b> checkbox.</p> <p>d. <b>Optional:</b> To speed up case creation, enable automatic creation of participants when the case starts by selecting the <b>Create participant automatically when the case starts</b> checkbox.</p>
<b>Participant without an account in your application</b>	<p>a. In the <b>Role has user account with this application</b> section, select <b>No</b>.</p> <p>b. <b>Optional:</b> To save time and create more participants that have the same notification settings, enable creation of multiple participants for this role by selecting the <b>Allow multiple participants for this role</b> checkbox.</p>

10. **Optional:** To process a case more efficiently, in the **Preferences** section, define additional settings for the role:
  - To save time and create more participants that have the same notification settings, enable creation of multiple participants for this role by selecting the **Allow multiple participants for this role** checkbox.
  - To speed up case creation, enable automatic creation of participants when the case starts by selecting the **Create participant automatically when the case starts** checkbox.
11. Click **Done**.
12. Click **Save**.

[Sending confirmations of completed assignments \(on page 239\)](#)

[Sending custom notifications about new assignments \(on page 239\)](#)

[Sending email notifications from cases \(on page 236\)](#)

[Sending push notifications about case events \(on page 238\)](#)

## Predicting case outcomes

Meet your business goals and optimize the way your application handles work by using predictions in your case types. You can add more than one prediction to a case type to predict various outcomes, such as the probability of a successful case resolution or the probability of fraud.

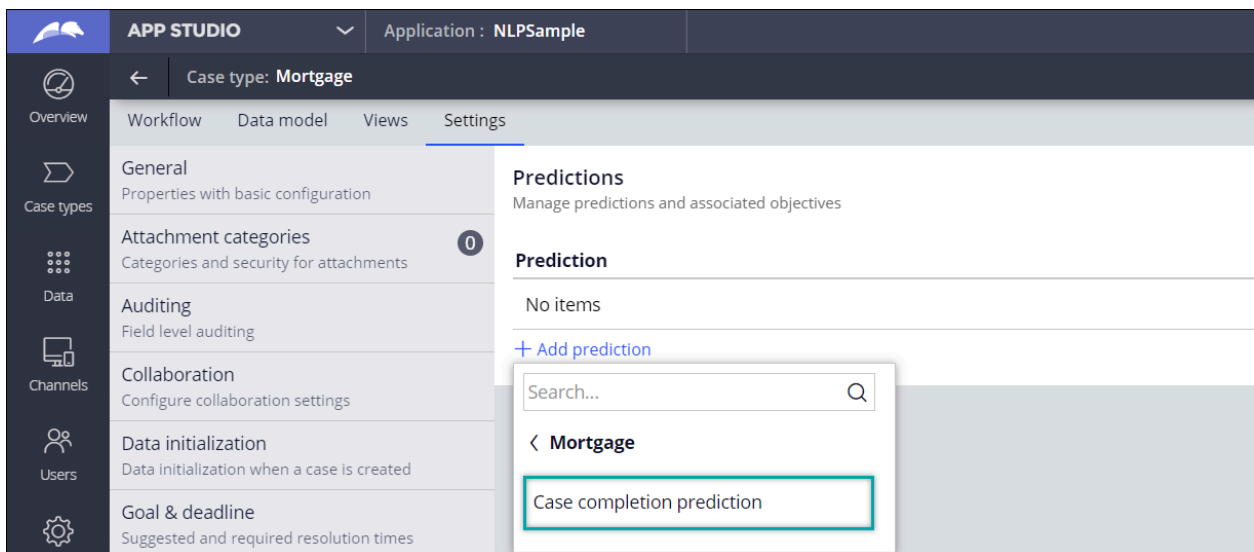
Predicting outcomes helps you manage your case work more effectively and achieve the following goals:



- Prioritize cases that are most likely to bring you value.
- Prioritize work that needs immediate attention.
- Route suspicious cases for closer inspection.

Make your application respond differently to individual cases based on the predictive fields that feed into predictions that result in the probability of an outcome. You can use predictions when you create a condition, by using the condition builder. Predictions can be useful when you add a decision to your process, or when you define the conditions for routing assignments. For example, your application can route an assignment to a high priority work queue when a customer has a high net worth and the prediction to successfully resolve a mortgage case is greater than 80%.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case work area, click the **Settings** tab.
4. In the navigation pane, click **Predictions**.
5. In the **Predictions** window, click **Add prediction**, expand the list of available predictions, and then select a prediction that you want to use.



6. Click **Save**.

[Adding decisions to processes \(on page 280\)](#)

[Assigning users automatically at run time with business logic \(on page 85\)](#)

[Displaying optional actions conditionally \(on page 355\)](#)

[Displaying supporting processes conditionally \(on page 358\)](#)

## Including indexed data during case search

Enable search for a case type to ensure that users can find cases based on any text, partial or full, that matches indexed case data.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Settings** tab, click **Search and reporting**.
4. Click **Enable search**.
5. **Optional:** To make reporting faster without putting additional requirements on the database, click **Generate dedicated index**.

---

Creating dedicated indexes (on page )

## Troubleshooting errors with email instantiation

During email instantiation, some issues might occur that prevent case creation and, as a result, hinder the workflow in your organization. By troubleshooting errors with email instantiations, you increase the automation of case processing and deliver efficient software. For example, an insurance agent can send an email to an application to create a case after the agent assesses vehicle damage at a car accident site. By troubleshooting any issues related to email instantiation, you ensure that the application correctly creates a new case, and that the workflow in your enterprise is uninterrupted.

### Condition

Users want to create a case by sending an email to a specified email account. An issue with the email account configured for email instantiation occurs and case creation fails.

### Unavailable email account

The email account that you configured for case instantiation is currently in use by another case type.

### Solution: Creating an email account exclusively for a case type

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **Email instantiation**.
3. Clear the **Enable email listener** check box.



- Find and manually delete the set of standard rules that your application created when you enabled email instantiation for your case type.

For a list of these rules, see [Standard rules used by email instantiation \(on page 247\)](#).

**CAUTION:** Do not delete the email account itself because another case type might rely on it.

- In the case type rule form header, click **Save**, and then click **Actions > Refresh**.

You disassociated the email account from your case type.

- Create a new email account exclusively for your case type.

For more information, see [Configuring outbound email in App Studio \(on page 226\)](#).

- Enable email instantiation for your case type by using your new email account.

For more information, see [Enabling creation by email of top-level cases \(on page 226\)](#).

## The email account cannot send or receive emails

The email account that you configured for email instantiation has issues when sending or receiving emails.

### Solution: Testing and debugging email account connectivity

- In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
- On the **Settings** tab, click **Email instantiation**.
- Below the **Email address** field, click the email address that you associated with the case type, as shown in the following figure:

Email account for case instantiation

- On the **Email account** tab of the email account rule form, test connectivity:

- If the email account cannot send emails, in the **Sender** section, click **Test connectivity**.
  - If the email account cannot receive emails, in the **Receiver** section, click **Test connectivity**.
5. In the **Test Connectivity** window, review the test results.
  6. **Optional:** To obtain more information for debugging, inspect the log files:
    - a. In the header of Dev Studio, click **Configure > System > Operations > Logs**.
    - b. On the **Logs** tab, in the **Log utilities** section, click **Log files**.
    - c. In the **Log File Download** window, examine the logs.  
For more information, see *Viewing logs (on page )*.

## Redundant HTML tags in incoming emails

Incoming email messages include redundant HTML tags that cause case creation by email to fail.

### Solution: Removing redundant HTML tags from an email text

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Technical** category, and then click **Activity**.
3. In the list of activity instances, open the **pyTreatIncomingText** activity.
4. In the rule form header, click **Save as**.
5. In the **Context** section, in the **Apply to** field, select a class that includes the case type that you configured for email instantiation.
6. Click **Create and open**.
7. In row 1. of the activity that includes a Java method, click **Expand to see method parameters**.
8. In the **Java Source** text box, edit the Java script to exclude HTML tags from the email text:
  - To remove tags related to script and script elements, include the `doc.select("script").remove();` line in the script.
  - To remove tags related to styling, include the `doc.select("style").remove();` line in the script.
  - To remove tags related to formatting, include the `doc.select("form").remove();` line in the script.
  - To remove line breaks, include the `doc.select("br").remove();` line in the script.
  - To remove metadata tags, include the `doc.select("meta").remove();` line in the script.
9. In the activity header, click **Save**.

## Your email account is flagged as suspicious

The email provider that you included in the email account configuration flagged your email account as suspicious.

1. Access your email account by using a web browser or supported client application.
2. Log in by entering your user name and password.
3. Follow any additional steps from your email provider to successfully access the account.

## Email listener does not respond

Email listener does not respond because the email messages are marked as read.

## Solution: Keep email messages unread

Do not open and read email messages because email listeners respond only to unread messages. Notify your development team of this restriction.

---

[Troubleshooting email listeners \(on page 247\)](#)

[Creating a Service Email rule \(on page 247\)](#)

[Integrating your application with an email provider \(on page 247\)](#)

[Creating an email listener \(on page 247\)](#)

[Email account limitations \(on page 227\)](#)

[Standard rules used by email instantiation \(on page 247\)](#)

## Standard rules used by email instantiation

When you enable email instantiation for a case type, a set of rules and data instances are created.

Although these rules are managed by your application, it is good to know how they are named and what functionality they provide in the event that you need to remove them or debug an issue.

The following standard rules are used by email instantiation:

- Email account — Contains email account credentials

The name of this data instance is provided by the person who created the email account. You cannot create an email account by updating the settings of a case type.

- Email listener — Watches the specified email account for new messages

The identifier for this rule has two parts:

- The word "Listener"
- The class of your case type, the name of your starting flow, and the word "Listener" separated by underscores
- Service email rule — Runs your starting flow to create a case

The identifier for this rule has three parts:

- The class of your case type without hyphens
- The class of your case type
- A label that concatenates the word "Create" and the name of your starting flow
- Service package — Manages services

The name of this rule is the class of your case type without hyphens.



**Note:** These standard rules are not displayed in the Application Explorer because they do not have an applies to class. You can find these rules by searching, opening the Records Explorer, or viewing the **Dev Studio > Integration > Email** landing page.

[Troubleshooting errors with email instantiation \(on page 244\)](#)

[Enabling creation by email of top-level cases \(on page 226\)](#)

## Configuring complex processes

Reach your business goals and solve cases that require additional processing by configuring advanced options for processes. For example, you can define different paths that a process can follow after a decision point. You can also create subprocesses to support additional work that your case requires.

Typically, you build processes in Case Designer. For advanced configuration, you can build a process by adding shapes to a flow diagram. If you build a process by using a flow diagram, for future edits you also need to edit the process by using the flow diagram.

## Shapes and connectors

When you work with the flow diagram, you first select flow shapes, and then connect the shapes by dragging connectors. Every shape represents an action or a set of closely related actions in a process, such as an assignment that a user completes at run time, or an automation that an application performs.





Connectors show the direction in which a case moves forward and define the order of actions in a process. You can label shapes and connectors for more straightforward communication of events in a case.

Compared to Case Designer, the flow diagram offers additional shapes and configurations. For example, you can create a subprocess that gathers related tasks, and then you can reuse the subprocess in different processes. As a result, you create granular and efficient software. For example, in a hiring application, you can create a subprocess for preparing an offer, and then reuse the subprocess in other processes that users use to review candidates for different positions in a company. You can also use a decision point that determines the path of the process based on the decision result. For example, in a recruitment process, the decision point determines actions that occur after accepting or rejecting a candidate. For more information about shapes, see [Flow shapes \(on page 251\)](#).

Pega Platform conforms to the Business Process Model and Notation (BPMN) standard by using a subset of BPMN, but also extends the standard with smart shapes such as the Capture signature, Cascading approval, and Wait shape. The flow diagram includes a smart shapes palette that provides a set of frequently used, pre-configured utilities and sub-processes. The shapes in the palette can be used in process flows with little or no additional configuration beyond what BPMN supports. Pega continues to define and redefine the BPM space by favoring a more robust approach that supports advanced enterprise BPM constructs, such as business rules and case management (for example, treating GUI calls as a service instead of as an intrinsic part of the BPM flow).

---

### Related information


[Processes in a case life cycle \(on page 66\)](#)

[Empowering knowledge workers \(on page 348\)](#)

## Creating a stand-alone process

Create a stand-alone process to support or supplement the steps in a case life cycle. By creating a process outside the context of a case life cycle, you can reuse the process in other case types. For example, in a banking application, you can create a process for collecting feedback, and then reuse the process in different case types, such as resolving a credit card dispute or opening a new bank account.

**Note:** By default, you create processes in Case Designer in a context of a case type, and then you can reuse this process only in this case type. You can create stand-alone processes only in Dev

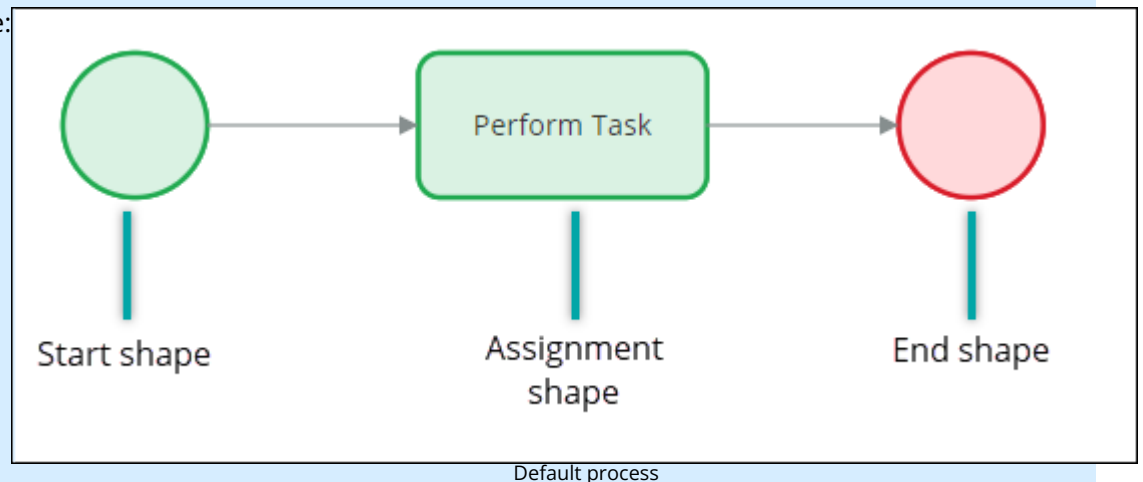
 Studio by using a flow diagram. If you want to use a flow diagram in App Studio, you need to add the process to a case life cycle first. For more information, see [Adding a sequential process to a stage \(on page 68\)](#).

1. Create a rule that stores the process:
  - a. In the header of Dev Studio, click **Create > Process > Flow**.

**Note:** A flow is a legacy name for a process.

- b. In the **Label** field, briefly describe the purpose of the process.
- c. **Optional:** To change the default identifier for the process, click **Edit**, and then provide a unique value in the **Identifier** field.
- d. In the **Context** section, select an application layer to store the process.
- e. In the **Apply to** field, click the Down arrow key, and then select a class that stores the process.
- f. In the **Add to ruleset** list, select a ruleset and a ruleset version to store the process.
- g. Click **Create and open**.

The flow diagram opens. By default, every new process includes the Start shape, one Assignment shape, and the End shape, as shown in the following figure:



2. On the **Diagram** tab, click **Flow Shapes**, and then select a shape that you want to add to the flow diagram.  
For more information about flow shapes, see [Flow shapes \(on page 251\)](#).
3. Connect the shapes by dragging connectors from one shape to another.



**Tip:** Connectors that work properly are green. If you cannot place a connector in a selected place, the connector is red.

4. Click **Save**.






## Flow shapes




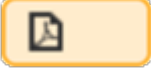
A shape in a process represents a task that a user or application completes as part of a business process. By connecting different types of shapes in a process, you can define the path that a case follows from creation to resolution.



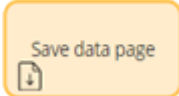


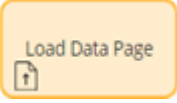

**Note:** Flow shapes of the `Automations` category are available only for processes that belong to a `Work-` class and its subclasses. If you configure a process that belongs to a `Data-` or `Rule-` class, the number of available automation shapes is restricted.



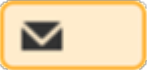


Name	Description	Image
Assignment	Creates a task in a work queue or worklist so that a user can provide input to the process. For example, you can collect income information and banking history from a customer in a loan request review process. For more information, see <a href="#">Collecting information from a user (on page 83)</a> .	
Decision	Evaluates an expression or calls a rule, such as a decision tree, to determine which step is next in the process progression. For example, in a loan request process, the case might follow multiple paths depending on the loan amount. see <a href="#">Adding decisions to processes (on page 280)</a> .	

Name	Description	Image
Subprocess	Calls another process from the current process. For example, in a hiring process, you can add a subprocess of running a background check. For more information, see <a href="#">Subprocesses (on page 287)</a> .	
Utility	Calls an activity or an automation. By using the Utility shape, you can perform processing, such as computations or data retrieval, without human input. For more information, see <a href="#">Calling an activity or automation from a process (on page 383)</a> .  <b>Note:</b> The Utility shape is available only in Dev Studio.	
Start	Designates the start of the process. For more information about the advanced configuration options, see steps from 3 <a href="#">(on page 277)</a> to 11 <a href="#">(on page 280)</a> in <a href="#">Creating a screen flow (on page 276)</a> .	
End	Designates the end of the process.	
Automations		
Attach content	Attaches a file or an URL to a case. You can select different types of files to attach. For example, in a hiring process, you can attach scanned university diplomas of a job candidate. For more information, see <a href="#">Attaching content to a case (on page 101)</a> .	




Name	Description	Image
Change to a stage	Moves a case to a specific stage that you select. For example, in a hiring process, you can move a case to an optional stage for an additional interview. For more information, see <a href="#">Moving a case to a different stage (on page 130)</a> .	
Change to next stage	Moves a case to the next stage. For example, in a hiring process, after a job candidate provides necessary documents, a stage moves to the next stage in which a human resources worker reviews the documents. By default, a case moves to the next stage after all the assignments in a current stage are complete. Use this shape when you want to move the case before completing some assignments. For more information, see <a href="#">Stages in a case life cycle (on page 46)</a> .	
Create case	Creates a top-level case or one or more child cases. For example, in a car accident insurance request review, you can create a child case to review vehicle damage. For more information, see <a href="#">Creating contextual cases (on page 106)</a> .	
Create PDF	Creates a PDF file from a specified section and attaches the file to the case. For example, in a car accident review request, you can create a PDF file with information about a vehicle damage. For more information, see <a href="#">Attaching screen captures to a case (on page 104)</a> .	



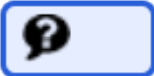

Name	Description	Image
Generate document	Generates a document that stores information from a case. You can select which fields that correspond with case data you want to include in the documents. For example, in a hiring process, you can generate a document that stores personal details of a candidate, the interview score, and a decision about accepting a candidate. For more information, see <a href="#">Generating case documents (on page 112)</a> .	
Persist case	Converts a temporary case to a permanent object in the database. For example, in a hotel booking system, an application creates a temporary case when a user logs in. If the user changes any booking details, the application persists the case to retain data. For more information, see <a href="#">Persisting temporary cases (on page 134)</a> .	
Save data page	Saves case information as a separate data page so that you can reuse data outside of a case. The information needs to come from a savable data page or an autopopulated property that points to the savable data page. After an update, an application saves the updated information to the same specified data page. For example, in an onboarding case, you can store information about a new hire, such as an office location or requested equipment, as a data page, and then reuse this information in other places of your application that require a data page. For more information, see <a href="#">Saving data</a>	

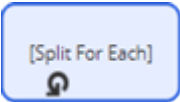

Name	Description	Image
	<p>in a data page as part of a case life cycle <i>(on page 148)</i>.</p> <p><b>Note:</b> The Save data page shape is available only in Dev Studio.</p>	
Load data page	<p>Preloads a data page with information that your application displays to users. As a result, case processing is faster, and users avoid waiting for an application to load data on demand. For example, in an onboarding case, an application can preload a data page that stores information about equipment that a new hire requests before a worker who processes the case needs to view the information. For more information, see <a href="#">Preloading a data page (on page 136)</a>.</p> <p><b>Note:</b> The Load data page shape is available only in Dev Studio.</p>	
Post to Pulse	<p>Creates a message that is sent to the Pulse social stream. For example, when a customer service representative (CSR) collects additional details from a customer, they can post a message to inform other stakeholders that the case involves. For more information, see <a href="#">Posting messages to a case (on page 135)</a>.</p>	




Name	Description	Image
Push notification	Sends a notification to an iOS or Android mobile device to indicate that a case requires a user action. For example, when a hiring process moves to a stage where a hiring manager needs to accept or reject a candidate, the manager receives a notification. For more information, see <a href="#">Sending push notifications from cases (on page 146)</a> .	
Run data transform	Calls a data transform to convert data from one format and class to another format and class. For example, while processing a purchase order, you can populate the shipping address with the data that the user provides as their billing address. For more information, see <a href="#">Calling a data transform from a case (on page 106)</a> .	
Send email	Sends an email to one or more work parties. For example, in a hiring process, a job candidate can receive an email with a summary of the process. For more information, see <a href="#">Sending automatic emails from cases (on page 142)</a> .	
Send notification	Sends a notification to users over one or more channels based on the configurations that you define. For example, in a loan request review, you can notify a customer when a case moves forward and changes its status. For more information, see <a href="#">Sending event notifications from cases (on page 140)</a> .	
Update a case	Updates a selected case or all child cases with details that you select. For example, at a certain point of a loan re-	

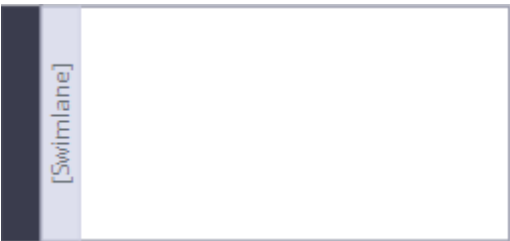






Name	Description	Image
	quest review, you can update the case urgency to a higher value to ensure that the case meets the resolution on time. For more information, see <a href="#">Updating case information (on page 149)</a> .	
Approval	Routes a case to one or more reviewers for an approval, based on a user name, reporting structure, or authority matrix. For example, in a hiring process, to hire a candidate for a managerial position, a case might require an approval from multiple managers on different levels. For more information, see <a href="#">Requesting approval from users (on page 91)</a> .	
Search duplicate cases	Returns a list of cases that are similar to a current case based on criteria that you provide. For example, if a customer creates a loan request similar to another loan request from the same customer that is still in progress, you can find and then close the second request. For more information, see <a href="#">Searching duplicate cases (on page 109)</a> .	
Wait	Pauses an assignment for a specified length of time or until one or more cases reach a specific status. For example, in a hiring process, you can pause the case until a human resources worker finishes a background check of a candidate. For more information, see <a href="#">Pausing and resuming processes in cases (on page 131)</a> .	

Name	Description	Image
Questionnaire	<p>Runs a questionnaire child case that asks users a set of questions in various formats. For example, you can provide a text box in which users can enter their answers directly, or create a radio button matrix, so that users can select their answers from a group of radio buttons. For more information, see <a href="#">Running a questionnaire in a case (on page 138)</a>.</p>	
Question	<p>Asks users a question in a format that your business process requires, such as plain text, a picklist, or a slider, and saves the answer in a case or system property. As a result, you can selectively capture user input without the formal structure of a questionnaire. For example, in an insurance process, a case worker asks a customer about the type of insurance claim that they want to make. For more information, see <a href="#">Asking a question in a case (on page 449)</a>.</p> <div data-bbox="423 1346 933 1480" style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p> <b>Note:</b> The Question shape is available only in Dev Studio.</p> </div>	
Question page	<p>Asks users a group of related questions in various formats that your business process requires, such as plain text, a picklist, or a slider, and saves the answers in case of system properties. For example, in a sales call, a telemarketer asks a customer a set of questions about their hobbies, and then uses the answers to provide on-</p>	

Name	Description	Image
	<p>ly relevant products. For more information, see <a href="#">Asking a group of related questions in a case (on page 451)</a>.</p> <div data-bbox="423 415 933 598" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 5px;"> <p><b>Note:</b> The Question page shape is available only in Dev Studio.</p> </div>	
Advanced shapes available only in Dev Studio		
Split for Each	<p>Creates multiple instances of the same subprocess that later rejoin the parent process. Users can work on the subprocesses simultaneously, which accelerates case resolution. For example, during the recruitment process of a job candidate, a hiring manager and an HR worker can review and assess the results of a job interview in parallel. For more information, see <a href="#">Running multiple instances of the same subprocess with a Split for Each shape (on page 299)</a>.</p>	 <p>The icon for the 'Split for Each' shape is a light blue rounded rectangle with a blue border. It contains the text '[Split For Each]' in a sans-serif font and a small circular icon with a downward-pointing arrow below it.</p>
Split Join	<p>Creates multiple subprocesses that later rejoin the parent process. Users can work on the subprocesses in parallel, which saves time during case resolution. For example, during the recruitment of a job candidate, an HR worker can conduct the background check while the hiring manager reviews and assesses the job history and certificates. For more information, see <a href="#">Running multiple instances of different subprocesses with a Split Join shape (on page 295)</a>.</p>	 <p>The icon for the 'Split Join' shape is a light blue rounded rectangle with a blue border. It contains the text '[Split Join]' in a sans-serif font and three vertical bars of varying heights below it.</p>

Name	Description	Image
Integrator	Identifies an activity that connects your instance of Pega Platform to an external system to send or receive data. The Integrator shape runs synchronously and moves a case to the next step after it receives the response from the external system, typically after a few seconds. For example, you can use the Integrator shape to retrieve customer account balances or verify account numbers in an external database. For more information, see <a href="#">Connecting a process to an external application (on page 380)</a> .	 <p>The image shows a yellow rounded rectangular icon with a black arrow pointing left. The text "[Integrator]" is centered above the arrow.</p>
Assignment Service	Identifies an activity that connects your instance of Pega Platform to an external system to send an assignment for external processing. The Assignment Service shape moves a case to the next step only after receiving the response from the external system when the assignment processing is complete. For example, an application can route an assignment to a third-party evaluator to estimate the value of vehicle damage in an insurance claim. For more information, see <a href="#">Assigning a task to an external application (on page 377)</a> .	 <p>The image shows a green rounded rectangular icon with a black arrow pointing left. The text "[Assignment Service]" is centered above the arrow.</p>
Assign to robot queue	Identifies the robotic work queue and automation to run on the work queue for process automations. As a result, you promote automation and minimize human engagement in the processes. For example, a robotic queue can automatically check currency rates for a banking application during international money transfers. For more informa-	 <p>The image shows a blue rounded rectangular icon with a black arrow pointing left. The text "Robot queue" is centered above the arrow. Below the text are two small icons: a robot head and a plus sign.</p>

Name	Description	Image
	tion, see <a href="#">Assigning a task to a robotic work queue (on page 374)</a> .	
Swimlane	Identifies a group of related shapes that are associated with the same organizational unit or business purpose. For example, after accepting a job candidate, an HR worker collects the necessary documents and prepares a legal agreement, while an IT worker prepares a work station and configures a laptop for the new worker. You can use a swimlane to categorize the assignments between the HR worker and the IT worker, which facilitates routing of work.	
Conversation		
Ask a question	Asks users a question in a button, quick reply, yes/no, or open format, and saves the answer in a case or system property. The shape can also use smart validation to ensure that the user response is in the correct format, for example, a date format. You can only add this shape to a conversation process. For example, in a conversation process related to a vehicle ride request cancellation fee dispute, you can configure the shape to ask the reason for disputing the cancellation charge using buttons with three choices: specifying that the ride never arrived, the ride was late, or another reason. For more information, see <a href="#">Adding a case type conversation process for a conversational channel (on page 374)</a> .	

Name	Description	Image
Send a message	Displays a text message to the user. You can only add this shape to a conversation process. For example, in a conversation process related to a vehicle ride request cancellation fee dispute, you can configure the shape to inform users that a charged cancellation fee will be refunded within the next few days. For more information, see Adding a case type conversation process for a conversational channel ( <i>on page</i> ).	
Conversation flow	Starts another existing conversation process within the context of the work item, or in an embedded page if you create a nested data model for your case using the object-oriented approach. For example, for a vehicle repair case, you can configure the system to start an additional conversation process within the context of the work item that only asks users a series of questions about the extent of their vehicle damage. For more information, see Adding a case type conversation process for a conversational channel ( <i>on page</i> ).	
Other		
Annotation	Adds notes or comments to the flow without affecting how the flow runs. For example, in a case of preparing a business offer for a potential business partner, case workers can add comments about possible modifications of the options that the offer includes. For	

Name	Description	Image
	more information, see <a href="#">Annotating a process (on page 275)</a> .	

[Case life cycle elements \(on page 43\)](#)

[Creating a stand-alone process \(on page 249\)](#)



[Running multiple instances of different subprocesses with a Split Join shape \(on page 295\)](#)





[Running multiple instances of the same subprocess with a Split for Each shape \(on page 299\)](#)

## Shape icons in a process




Shape icons in a process are a visual indicator of a shape's configuration and current processing. You can use shape icons to quickly review the functionality and progress of a process without having to open the individual shapes. For example, you can view which assignments have corresponding tickets, or if an assignment generates any notifications.

You can view shape icons on assignments when you edit your process in a flow diagram. Shapes in a process support the following icons:

Icon	Description
	<p>The assignment has one or more corresponding tickets. By creating a ticket, you can jump to a specific assignment in a process and skip a part of the processing. For example, a user can skip assignments to provide the personal details of their spouse if they select an option that they are single in one of the previous assignments.</p> <p>For more information, see <a href="#">Creating a ticket (on page 351)</a>.</p>
	<p>The assignment includes a service-level agreement (SLA). For example, you can define a deadline of three business days to resolve an assignment.</p> <p>For more information, see <a href="#">Setting service-level agreements (SLAs) for stages, processes, and steps (on page 330)</a>.</p>

Icon	Description
	<p>The assignment generates one or more notifications. For example, you can notify interested stakeholders about the progress of a case.</p> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p><b>Note:</b> You can add notifications directly to assignments only in Dev Studio. For more information, see <a href="#">Sending custom notifications about new assignments (on page 239)</a>.</p> </div>
	<p>The assignment includes routing configurations. For example, an application can route an assignment to a current user or a work queue that a group of users can access.</p> <p>For more information, see <a href="#">Assigning tasks to users (on page 84)</a>.</p>
	<p>A spin-off subprocess runs simultaneously with the current process. The current process can end without completing the spin-off subprocess first. For example, an office worker can work on a spin-off subprocess of granting a new hire a parking spot in parallel to a main onboarding process.</p> <p>For more information, see <a href="#">Adding additional configuration to a subprocess (on page 292)</a>.</p>
	<p>When a case reaches an assignment, the case status changes to a value that includes the <code>New</code> prefix. For example, when a customer submits a new case and the case status changes to <code>New-Ready</code>.</p>



Icon	Description
	For more information, see <a href="#">Changing case statuses (on page 318)</a> .
	<p>When a case reaches an assignment, the case status changes to a value that includes the <code>Open</code> prefix. For example, when an application routes a case to a user for processing and the case status changes to <code>Open-InProgress</code>.</p> <p>For more information, see <a href="#">Changing case statuses (on page 318)</a>.</p>
	<p>When a case reaches an assignment, the case status changes to a value that includes the <code>Pending</code> prefix. For example, when an application routes a case to a manager for approval and the case status changes to <code>Pending-Approval</code>.</p> <p>For more information, see <a href="#">Changing case statuses (on page 318)</a>.</p>
	<p>When a case reaches an assignment, the case status changes to a value that includes the <code>Resolved</code> prefix. For example, when processing of a loan request ends and results in granting of the requested money, the case status changes to <code>Resolved-LoanGranted</code>.</p> <p>For more information, see <a href="#">Changing case statuses (on page 318)</a>.</p>

[Flow shapes \(on page 251\)](#)

[Adjusting a service-level agreement for the Assignment Service shape \(on page 379\)](#)

[Responding to business exceptions in a flow \(on page 348\)](#)

[Calling one process from another process \(on page 292\)](#)

[Changing case statuses \(on page 318\)](#)

## Types of processes

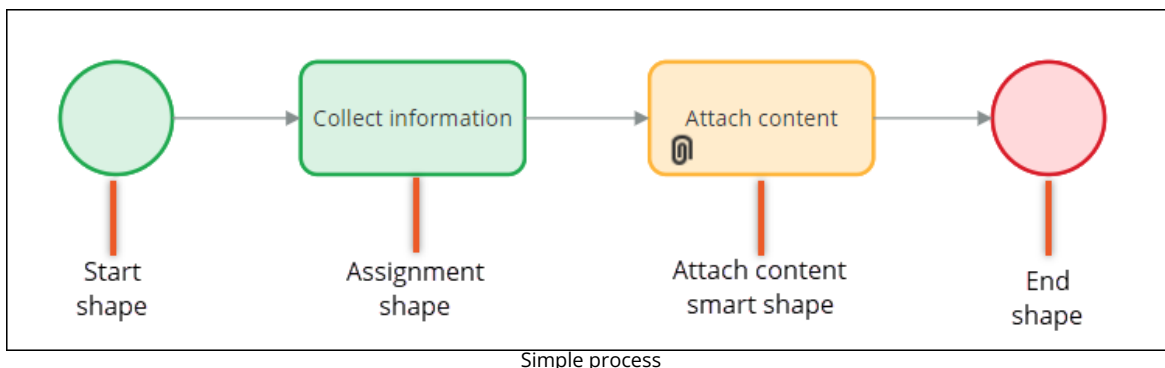
The default method of creating processes is by adding steps when you define a case life cycle. To access advanced options and configurations, you can also build your processes by using flow shapes on a flow diagram. You can create different types of processes based on the shapes that you choose or the way in which you integrate your process with the case life cycle. By understanding what distinguishes one process type from another, you can select the configuration options that meet your business needs. For example, in a case type for users to review car insurance claims, you can create parallel processes for reviewing vehicle damage and bodily injury. At run time, users complete the parallel processes simultaneously, which accelerates case resolution.

You can build the following types of processes:

### Simple

A process that contains a linear sequence of assignments, smart shapes, or subprocesses.

For example, a simple process can include an Assignment shape for collecting information from a job candidate and a smart shape to attach relevant documents to a case, as shown in the following figure:

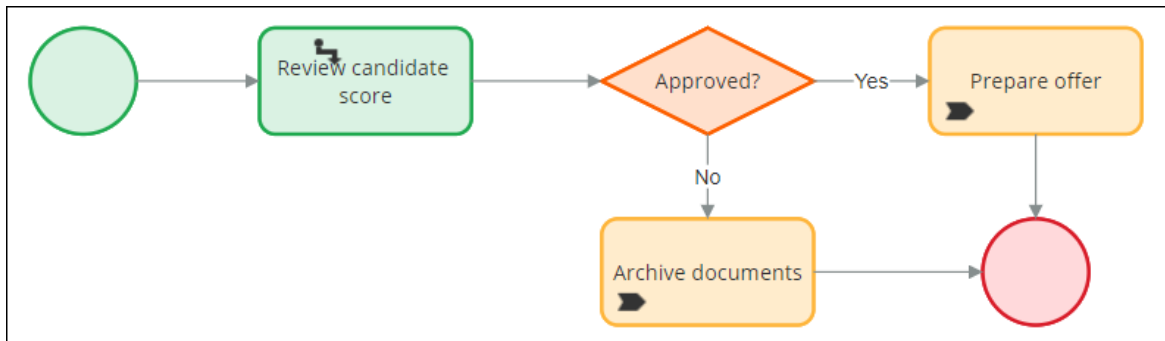


For more information, see [Creating a stand-alone process \(on page 249\)](#).

### Complex

A process that contains at least one alternate or conditional path. You define alternate paths by adding decisions to processes. For example, in a job candidate review process, you can define paths that the process follows after accepting or rejecting the candidate. You can also create a conditional path in a process, which runs only under specific circumstances. For example, you can trigger an additional background check if a job candidate applies for a managerial role.

The following figure shows a complex process that visualizes reviewing a job candidate. After a job candidate's score review, the process can follow two paths, based on the result of the review.



Complex process

For more information, see [Changing the path of a process \(on page 280\)](#).

### Straight-through

A process that contains only automations and does not require any human input. Straight-through processes can also apply to processes that contain only one human interaction or decision, or an assignment that users perform only in rare situations.

The following figure shows a straight-through process that includes using a robotic queue to check currency rates, apply currency rates through a data transform, and attach documents with the rates to a case. The process does not include any input or interaction from a user.



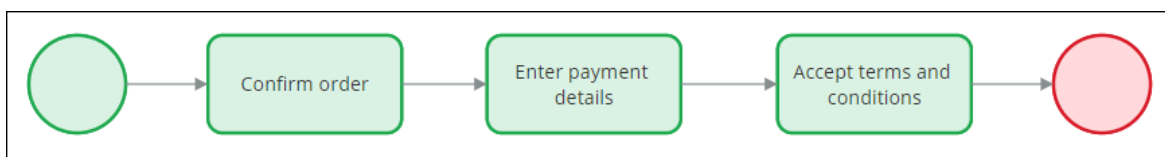
Straight-through process

For more information about automations, see [Flow shapes \(on page 251\)](#).

### Screen flow

A process that contains a Start shape that you configure to route assignments to a user or a work queue. In a screen flow, you configure routing only in the Start shape, so one user is responsible for completing the entire process. Users process a screen flow by interacting with assignments that the application displays on the screen. You can configure your application to display navigation between the assignments, for example as a breadcrumb trail.

For example, in a shopping application, you can create a screen flow that users complete to confirm their order, provide payment and shipment details, and review the terms and conditions of the shop, as shown in the following figure:



Screen flow

For more information about screen flows, see [Creating a screen flow \(on page 276\)](#).

In Case Designer, a counterpart of a screen flow is a multistep form. For more information, see [Adding a multistep form to a stage \(on page 70\)](#).

## Parent and subprocess

A process that calls another process by using a relevant shape is a parent. A process that a parent process calls is a subprocess. For example, in a parent process for reviewing vehicle damage in a car insurance request, you can call a subprocess that focuses on collecting visual materials and descriptions of the damaged car. By creating a hierarchy of processes, you build a more granular application, which is easier to maintain. You also promote reuse and manage the resources efficiently, because you can reuse one subprocess for multiple parent processes. For example, in a banking application you can reuse a subprocess for gathering feedback from customers by calling it from multiple processes, such as reviewing a loan request and resolving a credit card dispute.

You can use the following shapes to call subprocesses:

- Subprocess

For more information, see [Calling one process from another process \(on page 292\)](#).

- Split for Each

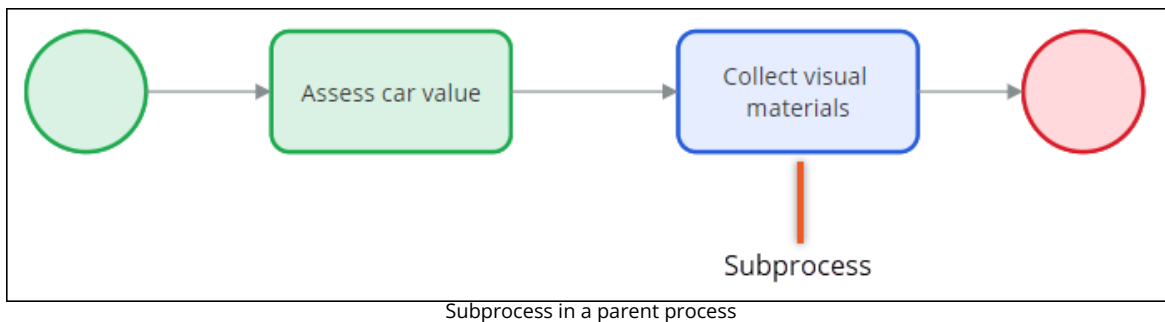
For more information, see [Running multiple instances of the same subprocess with a Split for Each shape \(on page 299\)](#).

- Split Join

For more information, see [Running multiple instances of different subprocesses with a Split Join shape \(on page 295\)](#).

By default, processing in a parent process resumes after all subprocesses reach their End shape.

The following figure shows a parent process for reviewing vehicle damage. The parent process includes a subprocess that users complete by submitting visual materials of the damaged vehicle:

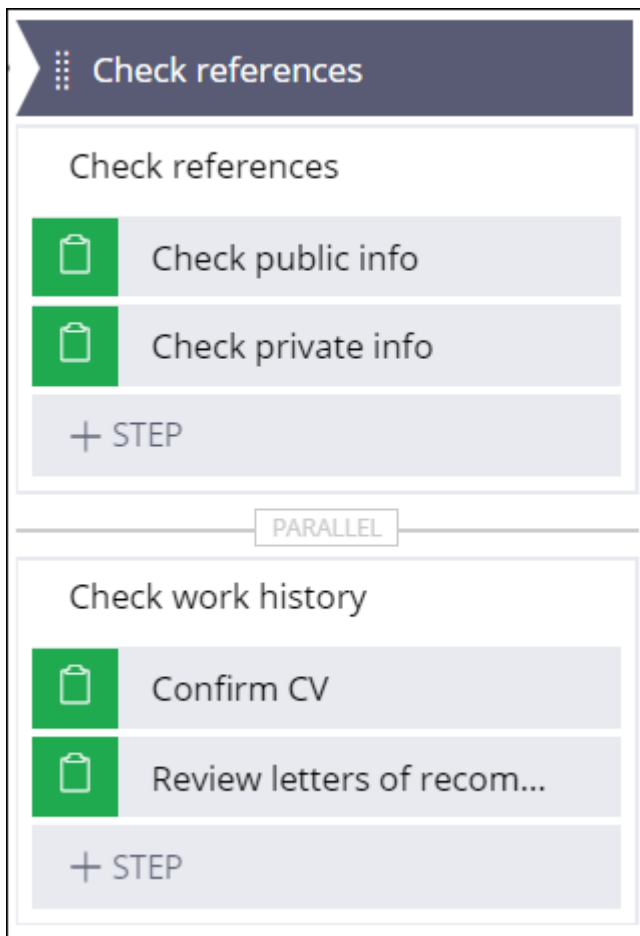


For more information, see [Subprocesses \(on page 287\)](#).

## Parallel

A process that runs asynchronously with a main process and allows multiple users to complete work simultaneously, which saves time during case resolution. For example, an HR worker performs a background check at the same time that a hiring manager reviews the work history and letters of recommendation of a job candidate.

Typically, you add parallel processes in Case Designer, as shown in the following figure:



Parallel processing

For more information about creating parallel processes in Case Designer, see [Adding a parallel process to a stage \(on page 69\)](#).

The Split for Each and Split Join shapes offer additional and more advanced options for parallel processing. You can also use a spinoff option on the Subprocess shape. For more information, see [Adding additional configuration to a subprocess \(on page 292\)](#).

For learning materials about parallel processing, see the [Parallel processing in Pega applications](#) module on Pega Academy.

---

[Creating a stand-alone process \(on page 249\)](#)

## Assignment shapes in processes

Assignment shapes represent tasks that users complete in a process. To ensure that your business process meets all your unique requirements, you can add an assignment shape for each place in a business process that requires human judgment and input. For example, you can add assignments to the life cycle of a case to collect information from users with different roles or levels of expertise, such as a form that a recruiter completes after interviewing a job candidate and reviewing the list of necessary documents in the recruitment process. You can also increase the functionality of your application by assigning tasks to an external application or to a robotic queue.

## Assignment shapes

You can create assignments by adding certain steps, such as the Collect information, Approve/Reject, or Form steps, to a process in Case Designer or by adding one of the assignment shapes to a process on the Diagram tab on the Flow form, which gives you more configuration options.

You can add, and then configure, the following assignment shapes:

### Assignment



The basic assignment shape that you use to collect data from users. The shape creates a task in a work queue or worklist, so that a user can provide input to the process. For example, you can use this shape in a mortgage case type to create a task for a case worker to calculate a new interest rate.

For more information, see [Flow shapes \(on page 251\)](#) and [Collecting information from a user \(on page 83\)](#).

**Note:** Some advanced settings for the Assignment shape are not available in Case Designer. You can access these settings on the Diagram tab by double-clicking an assignment in a flow diagram.

### Assignment Service



An advanced shape that you use to continue your work flow in an external system. The shape calls an activity that requests and receives information from an external system by using a connect rule. For example, you can use the shape to retrieve a customer's financial history from an external system or department.

For more information, see [Flow shapes \(on page 251\)](#) and [Assigning a task to an external application \(on page 377\)](#).

**Note:** Advanced shapes are available only in Dev Studio.

### Assign to robot queue



An advanced shape that you use to call a robotic automation. For example, you can use this shape to run an automation that analyzes the information submitted with an insurance claim.

For more information, see [Flow shapes \(on page 251\)](#) and [Assigning a task to a robotic work queue \(on page 374\)](#).

**Note:** Advanced shapes are available only in Dev Studio.

## Assignment settings

You can decide how users process the assignments in your application by configuring the following settings:

### Routing

Determines who processes the assignment. In a typical scenario, a regular assignment shape routes to a user (which in Dev Studio is known as the current or specific operator, or worklist), or a team (which in Dev Studio is known as a work queue, and formerly also known as workbasket).

For more information about the routing options, see [Collecting information from a user \(on page 83\)](#), [Assigning users automatically at run time with business logic \(on page 85\)](#), and [Configuring custom routing logic for assignments \(on page 89\)](#).

### The service-level agreement

Defines the recommended and required completion times for the assignment by creating goals and deadlines for case workers.

For more information, see [Completing work on time \(on page 324\)](#) and [Creating a service-level agreement \(SLA\) rule \(on page 332\)](#).

### Specification

Provides a description for the step. For more information, see [Creating a specification](#).

**Note:** To take advantage of the latest Pega Platform™ tools, use features instead of specifications. For more information, see [Adopting feature-driven development \(on page \)](#).

### Assignment configuration

Defines assignment details such as harness, confirmation note, and screen navigation.

For more information, see [Harnesses \(on page \)](#), [Sending confirmations of completed assignments \(on page 239\)](#), and [Creating navigation rules \(on page \)](#).

### Local actions

Specifies local actions that users can choose to take before they complete the assignment. At run time, local actions are displayed in the header of a case. To configure a local action for an assignment, you create or select an existing flow action. Submitting a local action at run time does not complete the current assignment.

For more information, see [About Flow Actions \(on page 302\)](#).

### Notifications

Sends out correspondence, such as an email message, when a running flow creates an assignment. Typically, the application sends this correspondence to one or more of the work parties identified in the work item, and then reports progress to that party. Users can select a notify activity, which is an activity of the notify activity type, on the Security tab.

For more information, see [Activity form - how to create activities for flows \(on page \)](#).

### Tickets





Determines whether an assignment is associated with a ticket, which can be raised at any time by any process in the same case. Processing is connected to a ticket to respond to a business exception, error flow, or event, such as for example a user withdrawal from processing.

For more information, see [Creating a ticket \(on page 351\)](#).

### Optimization

Records work item properties that you want to use in the optimization analysis with the Process Optimization tool, which performs probability analysis on assignments to identify common flow patterns and trends.

## Assignment types for custom routing options

Each assignment in a process has a type based on the standard activity that creates the assignment. You can select a standard activity when you define the routing options for an assignment in a flow diagram in Case Designer. When you select the **Custom** routing option on an assignment shape on the Diagram tab of the Flow form, you can then select an assignment activity from the list in the **Assignment Type** field:

### Connect

Creates an assignment that waits for an external system to perform processing before moving to the next step in the process.



**Note:** This assignment type supports the Assignment Service shape, which is available only in flow diagrams. For more information, see [Flow shapes \(on page 251\)](#) and [Assigning a task to an external application \(on page 377\)](#).

### Dependency

Creates an assignment that waits for an event to occur, such as the resolution of another case.



**Note:**



- This assignment type is available only in flow diagrams.
- As a best practice, use the standard Wait shape instead of a Dependency assignment type. For more information, see [Flow shapes \(on page 251\)](#) and [Pausing and resuming processes in cases \(on page 131\)](#).
- Use this assignment type only if you want to fine-tune the waiting logic with a when rule.

## External

Uses the Directed Web Access feature to create an external assignment, for example, to receive input from an external user.

For more information, see [Directed Web Access in configuring assignments for external users \(on page 387\)](#).



**Note:** This assignment type is available only in flow diagrams.

## Work queue

Creates an assignment that an application can route to a work queue that is common to multiple users.

For more information, see [Configuring custom routing logic for assignments \(on page 89\)](#), [Creating a work queue \(on page \)](#) and [Assigning users automatically at run time with business logic \(on page 85\)](#).

## Worklist

Creates an assignment that an application can route to the worklist of a current user or a specific user that the router activity defines.

For more information, see [Configuring custom routing logic for assignments \(on page 89\)](#) and [Assigning users automatically at run time with business logic \(on page 85\)](#).

---

[Collecting information from a user \(on page 83\)](#)

[Step types \(on page 79\)](#)

[Activity form - how to create activities for flows \(on page \)](#)

## Annotating a process

Promote sharing information with other members of your development team by annotating a process. When you document the functionality of a process, you can quickly communicate how a process works or fits business requirements, and consequently speed up application development and avoid unnecessary changes that might come from insufficient understanding of a functionality. For example, by analyzing annotations, new team members can quickly understand why certain actions occur in a process.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Workflow** tab, click **Life cycle**.
3. Hover over a process in a stage, and then click the **Configure process** icon.
4. On the toolbar, click the **Add a flow shape** icon, and then click **Annotation**.
5. Double-click the Annotation shape, and then replace the default text with your notes or comments.
6. **Optional:** To adjust the size of the Annotation shape to fit the text, resize the Annotation shape.
7. Drag the Annotation shape to a different position on the canvas, based on the functionality that you want to document.
8. Click **Save**.

At run time, annotations have no effect on processing.

[Changing the path of a process \(on page 280\)](#)

[Subprocesses \(on page 287\)](#)

## Keyboard shortcuts for processes

For faster and more efficient creation of processes, you can use keyboard shortcuts in the flow diagram. Using keyboard shortcuts also supports accessibility and provides a better experience for developers who rely on a keyboard while developing an application.

You can use the following keyboard shortcuts when you create a process in a flow diagram:

Shortcut	Description
Shift + mouse click	Selects an item or removes it from the rest of the selected items.  For example, you can select multiple flow shapes so that you can delete more than once.
Delete or Backspace	Deletes the selected items.
Ctrl + A	Selects all objects on the canvas.



Shortcut	Description
Alt	Temporarily switches between pan mode and select mode.
Secondary mouse button + drag	Temporarily pans the canvas when in select mode or temporarily selects items when in pan mode.
+ (Windows) or = (Mac OS)	Zooms in.
-	Zooms out.
Ctrl + M	Toggles between full screen and the normal canvas.
Esc	Exits the current operation.
Ctrl + S	Saves changes.
Double click on the canvas	Centers and fits the screen diagram.

Configuring an accessible UI (on page 276)

## Creating a screen flow

Create a screen flow to walk a single user through a series of screens to collect information. By dividing information into multiple screens, you provide a well-organized and user-friendly interface that is convenient to navigate. For example, in an application to book medical appointments, you can create separate screens for patients to enter their personal details, their medical history, and a description of their current issue. As a result, the patient interacts with logically organized information.



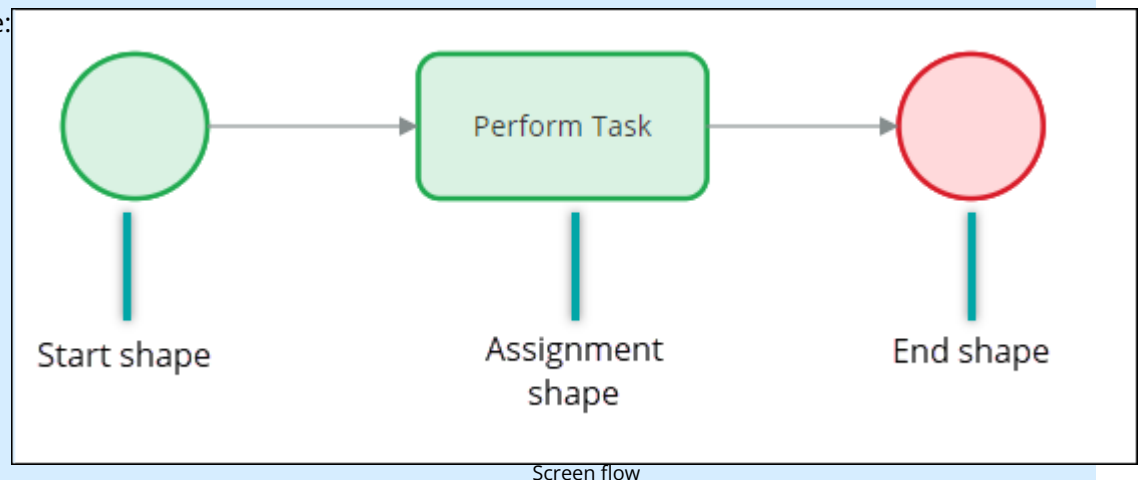
**Note:** Creating a screen flow in Dev Studio provides for advanced configurations and is suitable for advanced developers. To create a form with multiple screens by using low-code tools, switch to App Studio and create a multi-step form. For more information, see [Adding a multistep form to a stage \(on page 70\)](#).

Screen flow is a type of process. For more information, about processes, see [Types of processes \(on page 266\)](#).

1. Create the rule that stores the screen flow:
  - a. In the header of Dev Studio, click **Create > Process > Flow**.
  - b. In the **Label** field, describe briefly the purpose of the screen flow.

- c. **Optional:** To change the default identifier for the screen flow, click **Edit**, and then provide a unique value in the **Identifier** field.
- d. Click **View additional configuration options**, and then select **Standard template for screen flows**.
- e. In the **Context** section, select an application layer to store the screen flow.
- f. In the **Apply to** field, press the Down arrow key, and then select the class that defines the scope of the screen flow.  
The class controls which rules the screen flow can use, as well as which rules can call the screen flow.
- g. In the **Add to ruleset** field, select the name and version of the ruleset that stores the screen flow.
- h. Click **Create and open**.

A flow diagram with a new screen flow opens. By default, a new screen flow includes a Start shape, one Assignment shape, and an End shape, as shown in the following figure:



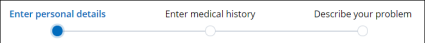
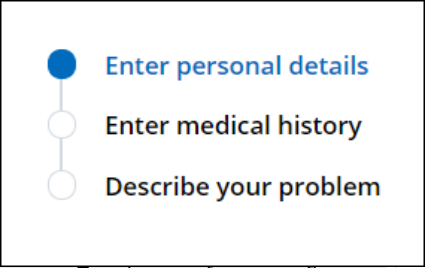
2. Open a dialog box with screen flow configuration by double-clicking the Start shape.
3. **Optional:** To provide a unique name for the Start shape, in the **Start Shape** field, enter a new value.
4. In the **Start** section, define how an application displays the screen flow:

#### Choices

**Apply a harness that you associate with a case type**

#### Actions

In the **Harness** list, select **Use case type policy**.

Choices	Actions
<p><b>Display navigation to screens as tabs</b></p>	<p>a. In the <b>Harness</b> list, select <b>Custom</b>.</p> <p>b. In the <b>Harness name</b> field, press the Down arrow key, and select <b>Tabbed-ScreenFlow7</b>.</p> <div data-bbox="938 445 1463 722" style="border: 1px solid #0070C0; padding: 5px;"> <p>At run time, an application displays assignments as horizontal tabs, as shown in the following figure:</p>  <p style="text-align: center;">Tabs in a screen flow</p> </div>
<p><b>Display navigation to screens as a tree view</b></p>	<p>a. In the <b>Harness</b> list, select <b>Custom</b>.</p> <p>b. In the <b>Harness name</b> field, press the Down arrow key, and select <b>TreeNavigation7</b>.</p> <div data-bbox="938 957 1500 1453" style="border: 1px solid #0070C0; padding: 5px;"> <p>At run time, an application displays assignments in a screen flow as a vertical tree layout, as shown in the following figure:</p>  <p style="text-align: center;">Tree layout of a screen flow</p> </div>
<p><b>Apply your custom harness</b></p>	<p>a. In the <b>Harness</b> list, select <b>Custom</b>.</p> <p>b. In the <b>Harness name</b> field, press the Down arrow key, and select your custom harness.</p>



**Note:** At run time, an application displays every assignment in a screen flow as a separate screen.

5. **Optional:** To save a case after the last step in a screen flow is complete, select the **Save on last step** checkbox.

If you leave the **Save on last step** check box clear, the case saves information each time a user completes an assignment. As a best practice, save the case data on the last step to avoid performance issues.

6. **Optional:** To allow users to move to the next assignments in a screen flow even if an assignment fails validation, select the **Allow errors** checkbox.



**Note:** An option to allow errors is available only after you configure the screen flow to save a case on the last step.

Users can switch between screens even if the information that they provide fails validation. However, users can submit the flow screen only after an entire flow passes validation on the last step.

7. In the **Status** section, in the **Work status** field, press the Down arrow key, and then select a status that a case obtains when the screen flow starts.

8. In the **Routing** section, in the **Route to** list, determine a user that completes the screen flow:

- To assign the screen flow to a user who currently processes the case, select **Current operator**.
- To assign the screen flow to another user in your application, select **Operator**, and then configure the user parameters.

You can assign the screen flow to a user by user name or user reference, the reporting manager of the user who currently processes the case, or a case participant.

- To assign the screen flow to a team that shares a work queue, select **Work queue**, and then select a team.

A user who picks up the assignment from a work queue needs to complete an entire screen flow.

- To assign the screen flow to a worklist or a team that shares a work queue, select **Custom**, and then in the **Router** field, enter a router activity to route the assignment.

For example, you can route a screen flow to an external work queue.

- To automatically determine a user at run time, select **Use business logic**, and then in the **Decision tree** field, enter a decision tree that returns a user to complete the screen flow.



**Note:** For a screen flow, you define routing only once, when you configure the Start shape. Routing configuration is unavailable on other flow shapes in the screen flow.

9. **Optional:** To define business requirements and instructions for the process, in the **Specification** section, associate a specification with your process:
- To use an existing specification, in the **Specification** field, enter the specification.
  - To create a new specification, enter text in the rich text editor.
  - To use a **.docx** file that stores a specification, click **Upload Word document as description**, and then in the dialog box, select the file that you want to use.

For more information, see [Creating a specification](#).

10. Click **Submit**.

11. Click **Save**.

---

[Collecting information from a user \(on page 83\)](#)

[Managing work across your team \(on page \)](#)

[Assigning tasks to users \(on page 84\)](#)

[Decision trees \(on page \)](#)

## Changing the path of a process

You can change the path of a process, or flow, to support out-of-sequence events. By adding different structures to your flow, such as loops and branches, you can improve the flexibility of the cases that you create.

The following tasks can help you change the path of a process:

---

[Creating a stand-alone process \(on page 249\)](#)

[Adding a conditional path to an assignment \(on page 285\)](#)

## Adding decisions to processes

Make your business processes flexible and responsive by providing conditional paths that a case can follow to reach a resolution. By adding a decision point to your case, you support more than one outcome for your business process, and adjust the case to changing business needs and circumstances.

For relevant training materials, see the [Automating workflow decisions](#) module on Pega Academy.





1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, on the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, hover over a process in a stage, and then click the **Configure process** icon.
5. Above the canvas, click the **Add a flow shape** icon, and then click **Decision**.

For advanced decisions that you cannot define in when conditions, configure the Decision shape to use a different logic type, such as a Boolean expression or a decision tree. For more information, see [Decision trees \(on page 281\)](#).

6. In the Decision shape, click **[Decision]**, and then replace the default label with a descriptive name.
7. Drag the Decision shape onto the canvas, to a position where the decision occurs in the process.
8. Connect the Decision shape to the process:
  - a. Drag a connector from a shape that precedes the decision in the process to the Decision shape.
  - b. Drag the end point of the connector from the Decision shape to the connection point of the shape that follows the decision on the canvas.

 **Tip:** The Decision shape works correctly when the connector is green.

9. Click the Decision shape.
10. In the Decision shape property panel, define at least one conditional path:

Choices	Actions
<p><b>Create a custom condition</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>When</b> list, select <b>Custom condition</b>.</li> <li>b. Click the <b>Configure conditions</b> icon.</li> <li>c. In the <b>Configure condition</b> window, select a condition, a comparator, and a value to compare with the condition.</li> <li>d. <b>Optional:</b> To define more conditions, click the <b>Add row</b> icon, and then repeat substep <a href="#">10.c (on page 281)</a>.</li> </ol>

Choices	Actions
	<p>e. <b>Optional:</b> To group the conditions, select comparators from the list.</p> <p>f. <b>Optional:</b> To reuse the condition in the future, click <b>Actions &gt; Add to when conditions library</b>.</p> <p>g. Click <b>Submit</b>.</p>
<b>Use an existing condition</b>	<p>a. In the <b>When</b> list, select <b>Existing condition</b>.</p> <p>b. In the list of conditions, select a condition.</p>

11. In the **Go to** list, select the shape to which the process advances when the case meets the conditions that you define.
12. In the **Otherwise go to** list, select the alternative shape to which the process advances when the case does not meet the conditions.
13. Click **Save**.

At run time, when the case reaches a decision point, the system evaluates the conditions. If the case meets the conditions, the process moves to the step that you defined. If the case does not meet the conditions, the system moves the case to the alternative step.

[Flow shapes \(on page 251\)](#)

[Adding a parallel process to a stage \(on page 69\)](#)

[Conditionally starting a process \(on page 74\)](#)

## Types of decision logic

You can use different types of logic when you add the Decision shape to a flow. By understanding what makes each type unique, you can create flows that make decisions more effectively.

You can use the following types of decision logic in a flow:

- Boolean expression - Returns a true or false value, based on an evaluation of conditions and properties that you define.
- Decision table - Returns a single value for the row in the table that meets the conditions, expressions, and value ranges that you define.



- Decision tree - Returns a single value for the branch in the tree that meets the if-then logic that you define.
- Fork - Evaluates when conditions, expressions, and likelihoods that you define to determine which connector advances the flow to the next shape in the sequence.
- Map value - Converts one or two input values into a single-value result.
- Predictive model - emailed Piotr and Bartek
- Scorecard Model - emailed Piotr and Bartek

---

[Adding decisions to processes \(on page 280\)](#)

[Changing the decision logic in a process \(on page 283\)](#)

## Changing the decision logic in a process

You can change the way that you implement a decision in a process. By using a rule instead of a series of when conditions to define your decision logic, you can create cases that support complex decisions and are easier to maintain.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. Click a process that contains a Decision step, and then click **Open process**.
5. On the **Diagram** tab of the Flow form, double-click the Decision shape to open the property panel.
6. In the **Type** list, select a type of decision logic.

For more information about the supported types, see [Types of decision logic \(on page 282\)](#).

7. Associate your decision logic with the Decision shape.

- For decision logic that you define in a decision table, decision tree, map value, predictive model, or scorecard:
  - In the **Rule** field, press the Down Arrow key and select the rule that defines your decision logic.
  - Provide input values, based on the type of rule that you select.



- **Input** - Passes text or a property reference to a decision tree.
  - **Row input** - Passes text or a property reference to a row in a map value rule.
  - **Column input** - Passes text or a property reference to a column in a two-dimensional map value rule.
- In the **Result** in field, enter the name of a property that stores the result of the decision.
- For decision logic that you define in a Boolean expression:
    - In the **Expression** field, enter a Boolean expression that defines your decision logic.
    - To access helpful tips and built-in functions for your expression, click the **Build an expression** icon.
  - For decision in a fork. Percentage — Follows the connector for a specific percentage of flow runs.

For example, you can ensure that 20 percent of mortgage applications take a different path in the flow for auditing purposes.

This option is available only when you connect a Decision shape that is set to type Fork .

8. Click **Submit** to close the property panel.

9. Click **Save**.

---

[Adding decisions to processes \(on page 280\)](#)

Decision tables (on page )

Decision trees (on page )

Scorecard rules (on page )

About Predictive Model rules (on page )

Map Values (on page )

Building expressions with the Expression Builder (on page )

## Adding a conditional path to a process

You can use a connector to add a conditional path to a process. By defining the events that cause your process to follow different paths, you can create cases that support more than one outcome.

1. Open a process by searching for it or by using the Application Explorer. For more information, see [Finding rules by class \(on page 250\)](#).
2. Define a new path by connecting a sequence of shapes to an existing shape in the process. For more information about the different shapes that you can use, see [Flow shapes \(on page 251\)](#).
3. Configure the conditions that cause the process to follow the new path.
  - a. Double-click the connector that goes from the existing shape in the process to the first shape in the new path.
  - b. In the Condition type list, select **When**.
  - c. In the When field, enter a when condition that your application evaluates at run time.
  - d. In the Likelihood field, enter a number between 1 and 100 to control the order in which your application evaluates the connector at run time.
  - e. Click **Submit**.
4. Review the configuration of all other paths, or connectors, from the shape to ensure that it supports the following best practices:
  - There is only one **Else** connector, so that the flow can continue when no conditions return a true value.
  - There are no **Always** connectors, which are applicable only when there is a single path from one shape to another.
  - Each connector provides a unique Likelihood value.
5. Click **Save**.

---

[Changing the path of a process \(on page 280\)](#)

[Adding a results-based path to a flow \(on page 286\)](#)

[Adding decisions to processes \(on page 280\)](#)

## Adding a conditional path to an assignment

You can use a connector to add a conditional path to an assignment. By defining the events that cause your flow to follow different paths, you can create cases that support more than one outcome. Your application assigns a task, or flow action, to a user only when it cannot find a connector on the Assignment shape with conditions that return a true value.

1. Open a process by searching for it or by using the Application Explorer. For more information, see [Finding rules by class \(on page 250\)](#).
2. Configure the current connector.
  - a. On the **Diagram** tab, double-click the connector that leads from the assignment to the next shape in the flow.
  - b. In the **Likelihood** field, enter an integer between 1 and 100 to control which connector the flow follows when there are no conditions that return a true value.
  - c. Click **Submit**.



3. Define a new path in the flow by creating a sequence of shapes and connecting it to the assignment.  
For more information about the different shapes that you can use, see [Flow shapes \(on page 251\)](#).
4. Configure the new connector.
  - a. Double-click the connector that leads from the assignment to the new path.
  - b. In the **Flow action** field, press the Down Arrow key and select a flow action that users process when the flow follows this connector because there are no conditions that return a true value.
  - c. In the **Likelihood** field, enter an integer between 1 and 100 to control which connector the flow follows when there are no conditions that return a true value.
  - d. Click **Submit**.
5. Define the conditions that cause the flow to follow the new path when it reaches the Assignment shape.
  - a. Double-click the assignment to open the **Assignment properties** dialog box.
  - b. In the **Advanced** section, click **Assignment details** , and then select the **Allow auto-process** check box.
  - c. Click **+ Add auto-process condition**.
  - d. In the **Perform action** field, press the Down Arrow key and select the name of a flow action for the new connector.
  - e. In the **When** field, press the Down Arrow key and select a when condition that controls whether the flow follows the new connector. At run time, when conditions are evaluated based on the list order that you provide.
  - f. **Optional:** To update case information after the flow action is complete, open the flow action and define a postprocessing activity or data transform.
6. Click **Submit**.
7. Click **Save**.

---

[Changing the path of a process \(on page 280\)](#)


[Collecting information from a user \(on page 83\)](#)

[Assignment shapes in processes \(on page 270\)](#)

## Adding a results-based path to a flow

You can use a connector to add a results-based path to a flow. By using run-time values to determine which path a flow follows, you can create cases that support more than one outcome.

**Note:** Use results-based paths for decisions that your application can make without human input.

-  For more information supporting decisions that users can make, see [Adding an alternate path to an assignment \(on page 285\)](#).

1. Open a process by searching for it or by using the Application Explorer. For more information, see [Finding rules by class \(on page 249\)](#).
2. Define a new path by connecting a sequence of shapes to an existing shape in the flow. For more information about the different shapes that you can use, see [Flow shapes \(on page 251\)](#).
3. Configure the result, or return value, that causes the flow to follow the new path.
  - a. Double-click the connector that goes from the existing shape in the flow to the first shape in the new path.
  - b. In the Condition type list, select Result .
  - c. In the Result field, enter text or a predefined value, based on the shapes that you are connecting. For example, a Decision shape returns True or False values when it is configured with a Boolean Expression, or property values when it is configured with a Decision Tree.
  - d. In the Likelihood field, enter a number between 1 and 100 to control the order in which your application evaluates the connector at run time.
  - e. Click **Submit**.
4. Review the configuration of all other paths, or connectors, from the shape to ensure that it supports the following best practices:
  - There is only one **Else** connector, so that the flow can continue when no conditions return a true value.
  - There are no **Always** connectors, which are applicable only when there is a single path from one shape to another.
  - Each connector provides a unique Likelihood value.

---

[Creating a stand-alone process \(on page 249\)](#)

[Creating a screen flow \(on page 276\)](#)

[Creating views for case types \(on page 173\)](#)

## Subprocesses

A subprocess is a process that starts after another process calls it. To create a subprocess, add a specific step to a process in the life cycle of a case or add specific shapes to a flow diagram. Subprocesses modularize the events in the life cycle of a case and increase reusability of your resources, as you can implement the same subprocess in multiple scenarios. In a sample scenario, you can run a subprocess of collecting user feedback in multiple case types, such as reviewing a loan request, resolving a credit card dispute, and processing an insurance claim. Additionally, subprocesses can reduce case resolution time because they allow for processing of case events in parallel. For example, in a case of preparing onboarding for a new hire, you can create subprocesses for an IT setup and facilities setup. Different workers can move each process forward, independently and in parallel, which leads to faster case resolution.



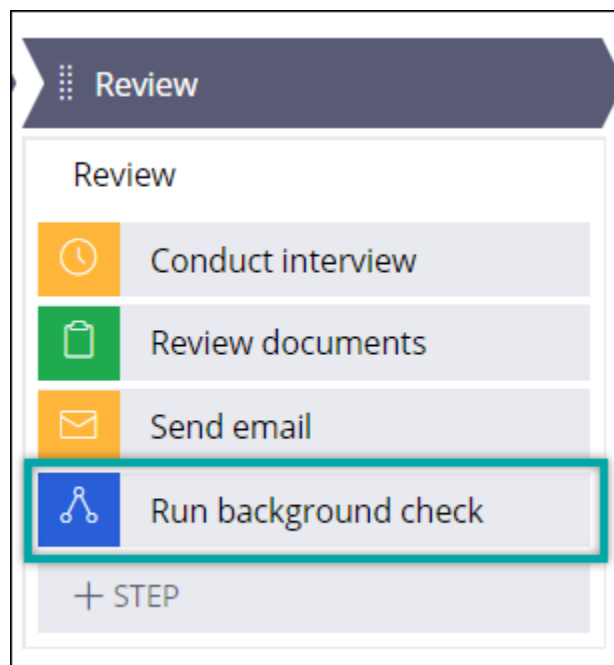
If you modify this topic, ensure that the companion "mini" topic is also updated. See: [Subprocesses \(on page 287\)](#)

You can configure different types of subprocesses to meet your business requirements. When implementing subprocesses into your case types, you have following options:

## Process step

In Case Designer, you can add a step that represents an existing process in your application. A case moves forward after the subprocess finishes. For example, in a Review stage of a Hire a job candidate case type, you can add a subprocess that represents running a background check. Internally, the subprocess consists of a series of steps, such as checking the criminal record and social security. As a result, you can maintain all the common business functionality with the subprocess for easier updates and clarity. Also, adding an entire process instead of single steps saves time and promotes the reuse of your application resources.

The following figure shows a Process step in a Review stage:



*Process step*

For more information, see [Calling one process from another process \(on page 292\)](#).

## Subprocess shape

In a flow diagram, you can add a Subprocess shape




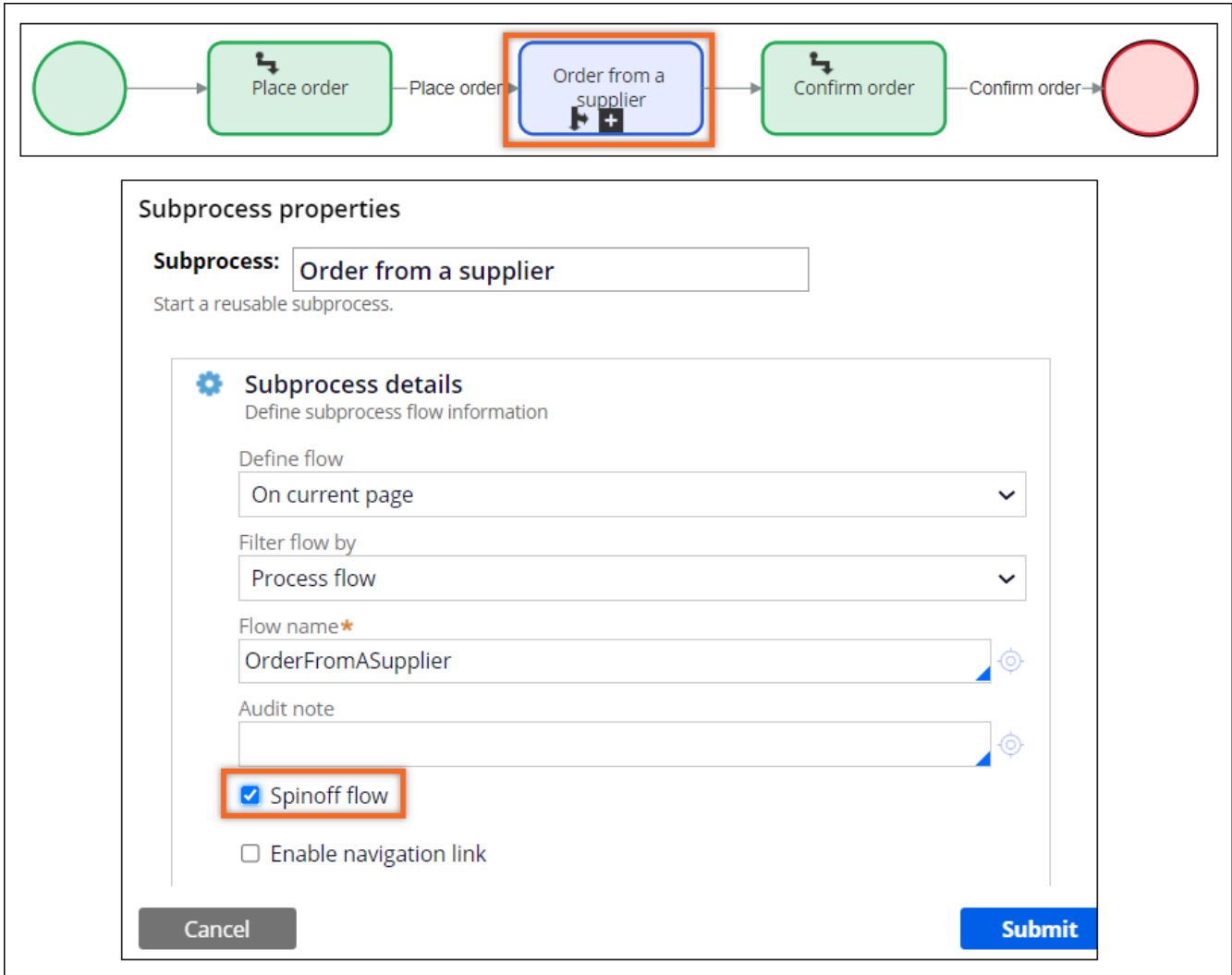
to a process in your case type. In the

background, a Subprocess shape is the same entity as a Process step that you add through Case Designer,



and both options have the same run-time effect. However, by adding a Subprocess shape through a flow diagram, you have more configuration options than a Process step. Additionally, in Dev Studio, you can configure advanced options for the Subprocess shape, such as defining a working context for a case. You can also start a subprocess as a spin-off. As a result, the subprocess runs independently from the main case, and the case can move forward without waiting for the subprocess to finish. For example, in an `Items purchase` case type, if the customer wants to buy more items than are currently in stock, an application can start a spin-off subprocess to order more items from a supplier. The subprocess runs independently, and the `Items purchase` case can reach resolution without waiting for the subprocess to rejoin.

The following figure shows a flow diagram with a process that includes an `Order from a supplier` subprocess. The arrows  icon indicates that the subprocess is a spin-off. The image includes also a dialog box with subprocess configuration.



The figure illustrates a flow diagram and its configuration. The flow diagram at the top shows a sequence of steps: a start node (green circle) leads to a 'Place order' step (green rounded rectangle), which leads to an 'Order from a supplier' subprocess (blue rounded rectangle with a spin-off icon), which leads to a 'Confirm order' step (green rounded rectangle), and finally to an end node (red circle). The 'Order from a supplier' subprocess is highlighted with an orange border.

Below the flow diagram is a 'Subprocess properties' dialog box. The 'Subprocess' field is set to 'Order from a supplier'. Under 'Subprocess details', the 'Define flow' dropdown is set to 'On current page', and the 'Filter flow by' dropdown is set to 'Process flow'. The 'Flow name' field contains 'OrderFromASupplier'. The 'Spinoff flow' checkbox is checked and highlighted with an orange border. The 'Enable navigation link' checkbox is unchecked. The dialog box has 'Cancel' and 'Submit' buttons.

*Spin-off subprocess*

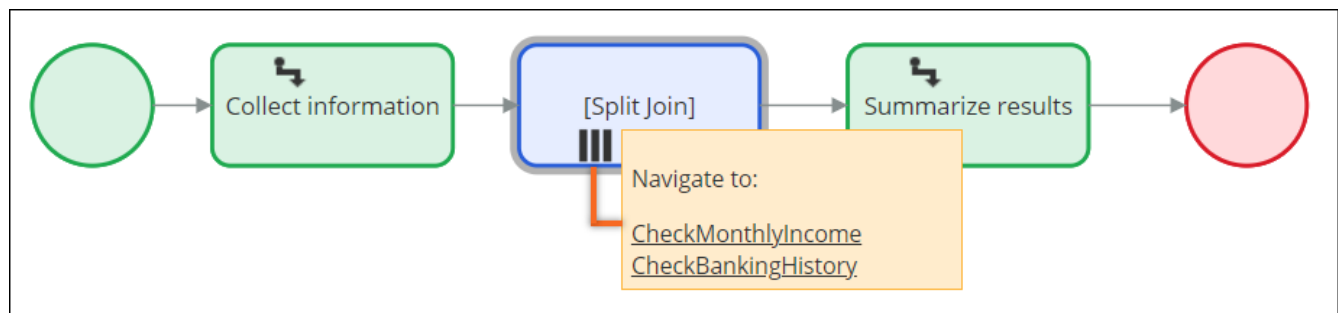
For more information, see [Adding additional configuration to a subprocess \(on page 292\)](#).

## Split Join shape

In a flow diagram in Dev Studio, you can add a Split Join shape  that starts multiple instances

of different subprocesses that later rejoin the main process. Based on the configuration that you provide, a case moves forward when one or more subprocesses rejoin the main process. For example, in a case of reviewing a loan request, you can use a Split Join shape to start subprocesses of reviewing a monthly income and banking history of an applicant. Different case workers can work on those subprocesses in parallel, which saves time.

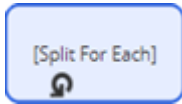
The following image shows a flow diagram with a process that includes the Split Join shape. The Split Join shape calls two subprocesses: `Check monthly income` and `Check banking history`.



*Split Join shape in a process*

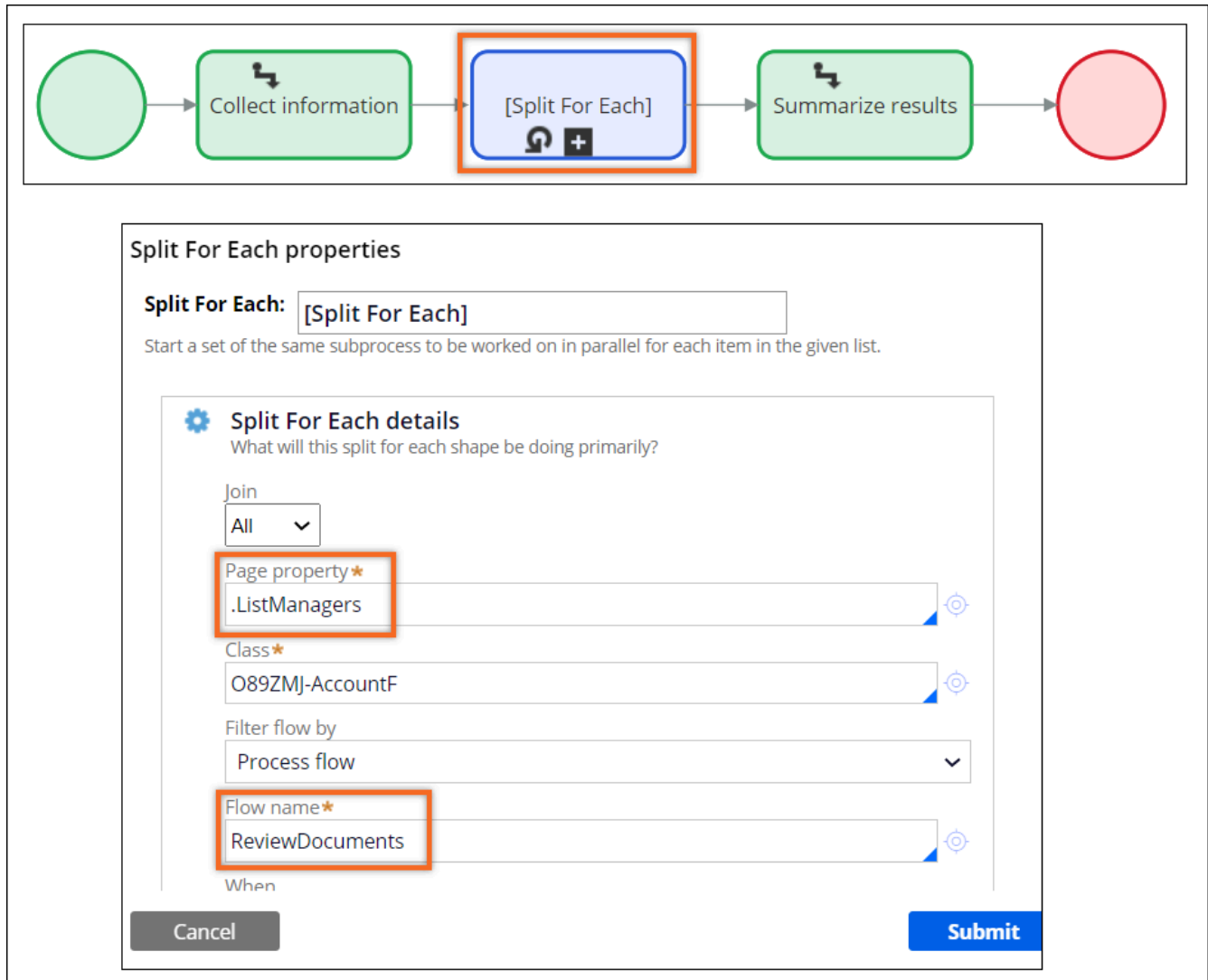
For more information, see [Running multiple instances of different subprocesses with a Split Join shape \(on page 295\)](#).

## A Split for Each shape

In a flow diagram in Dev Studio, you can add a Split for Each shape  that runs multiple

instances of the same subprocess. Based on the configuration that you provide, a main process resumes after one or more subprocesses rejoin the main process. You can also iterate a subprocess through all items on a page list that you provide, such as a list of managers. For example, in a `Hire a new candidate` case type, you can use a Split for Each shape to route a `Review documents` subprocess to multiple managers that need to review documents from a job candidate. The Split for Each shape routes work dynamically by using a page list, so in one instance of the `Hire a new candidate` case three managers might need to complete the `Review documents`, while in another instance the approval is required only from two managers, as the Split for Each shape generates the `Review documents` subprocess for each manager added to the case. Configuration options determine if all managers, any, or some need to approve before the case can proceed. The managers can review documents and complete the subprocess in parallel.

The following figure shows a flow diagram with a process that includes the Split for Each shape and a dialog box with a shape configuration. The Split for Each shape routes a `Review documents` subprocess to all managers that are listed in the `.ListManagers` page list. The system populates the page list dynamically for each instance of a case at run time.



*Split for Each shape in a process*

For more information, see [Running multiple instances of the same subprocess with a Split for Each shape](#) (on page 299).

[Processes in a case life cycle](#) (on page 66)

[Adding a sequential process to a stage](#) (on page 68)

[Adding a parallel process to a stage](#) (on page 69)

[Creating a stand-alone process](#) (on page 249)

## Calling one process from another process

Make your business processes more granular by complementing a process with a subprocess in your case life cycle. As a result, you avoid creating long and complex case types and deliver an application that is easier to maintain and edit if your business requirements change. Additionally, you save time because you reuse assets across your application. For example, in a loan request case type, you can create a process that represents revising a banking history of a customer, and then complement it with a subprocess of revising monthly regular payments. You can also reuse the subprocess for other case types, such as a mortgage request.

When you add a subprocess to your case life cycle, you add a whole set of steps at once. Consequently, you save even more time, because you avoid adding and configuring individual steps.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, locate the process that you want to complement with a subprocess, and then click **Step > More > Processes**.

**Tip:** You can adjust the list of subprocesses by managing relevant records in your application. For more information, see [Adding a relevant record to a specified class in your application \(on page 287\)](#).

4. **Optional:** To adjust the subprocess name to your current business needs, in the subprocess step, in the field, enter a new name.
5. Click **Save**.

---

[Subprocesses \(on page 287\)](#)

[Adding additional configuration to a subprocess \(on page 292\)](#)

[Creating a stand-alone process \(on page 249\)](#)

## Adding additional configuration to a subprocess

Provide access to relevant resources, such as properties or rules, that are outside your current case type by changing the scope of a subprocess in the case type. As a result, you ensure that users can access information that is necessary to successfully resolve their business processes even if the resolution requires data from another case. For example, a customer service representative (CSR) can resolve a complaint about hotel service by using information that a customer provided while booking a hotel room.



1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, in the header of a process that contains a subprocess, click **Configure process**.
3. On the toolbar, click **Open process**.
4. On the **Diagram** tab of the flow form, open the property dialog box by double-clicking the Subprocess shape.
5. In the **Subprocess details** section, in the **Define flow** field, select a working context for the subprocess:

Choices	Actions
<p><b>The subprocess works on a current case</b></p> <p><b>The subprocess works on a specific case</b></p>	<p>Select <b>On current page</b>.</p> <ol style="list-style-type: none"> <li>a. Select <b>On specific work item</b>.</li> <li>b. In the <b>Work property</b> field, enter a property reference in the current work item that identifies the key of the subprocess work item.</li> <li>c. In the <b>Class</b> field, enter the class of the subprocess work item.</li> <li>d. <b>Optional:</b> To specify a clipboard page that holds the subprocess work item, in the <b>Page name</b>, enter the clipboard page name.</li> </ol> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p><b>Note:</b></p> <p>If the case is open as a page on the clipboard, you can leave the <b>Work property</b> field blank.</p> <p>If the case does not already have a page and you do not provide a page name, your ap-</p> </div>

Choices	Actions
<p><b>The subprocess works on an embedded page</b></p>	<div data-bbox="850 218 1468 403" style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p>plication creates a page named <code>pyNextObj</code>, <code>pyNextObj_1</code>, and so on.</p> </div> <ol style="list-style-type: none"> <li>a. Select <b>On embedded page</b>.</li> <li>b. In the <b>Page property</b> field, enter a reference to the property of mode Page, Page List, or Page Group that holds the embedded page.</li> <li>c. In the <b>Class</b> field, enter the class of the embedded page.</li> </ol>

6. In the **Flow name** field, enter a name of the process that includes the subprocess for which you provide additional configurations.

By default, the system autopopulates this field with the name of the current process.

7. **Optional:** To display a message in the audit trail of the subprocess, in the **Audit note** field, enter a rule that stores the audit note that you want to include.

8. **Optional:** To allow the parent flow to continue processing without waiting for this subprocess to return, select the **Spinoff flow** check box.

When a case reaches the subprocess that runs as a spin-off, the subprocess runs asynchronously and independently from the main process. The main process can continue without waiting for the results from the subprocess. The main process and the subprocess might operate on the same work item or different work items.

9. **Optional:** To provide a link to the subprocess in the breadcrumb trail navigation, select the **Enable navigation link** check box, and then configure additional navigation options:

- To enable users to get back to the subprocess after the case moves forward, select the **Only allow navigating back to this step** check box.
- To enable post-processing or validation when users navigate back from the subprocess in the breadcrumb trail, select the **Perform post-processing when navigating away from step** check box.
- To allow users to return to the entry points in the subprocess, select the **Subprocess has navigation links** check box.

10. Click **Submit**.

11. Click **Save**.

[Calling one process from another process \(on page 292\)](#)

[Subprocesses \(on page 287\)](#)

## Running multiple instances of different subprocesses with a Split Join shape

Divide your independent business requirements into multiple processes by calling two or more subprocesses that later rejoin the parent process. As a result, you develop a granular and flexible application that you can conveniently adjust to your changing business requirements instead of creating long and complicated case types. For example, you can validate tax information and perform a title search as part of a mortgage application process. Different users can work on the assignments in each subprocess, and the parent process continues when one or both subprocesses return.



**Note:** Each subprocess runs asynchronously and in parallel. When a user works on an assignment in one subprocess, other users can work on an assignment from a different subprocess.

1. Add the **Split Join** shape to your process:
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the **Case life cycle** section, in the process in which you want to add the Split Join shape, click **Configure process**.
  - c. On the toolbar, click **Open process**.
  - d. On the **Diagram** tab, click the **Flow Shapes** icon, and then select **Advanced Shapes > Split Join**.
  - e. Connect the **Split Join** shape by dragging connector end points to connection points on different shapes in the process.  
Connectors that work correctly are green.
2. Open the dialog box by double-clicking the **Split Join** shape.
3. **Optional:** To provide a unique name for the shape, in the **Split Join** dialog box, in the **Split Join** field, enter a new name.
4. In the **Join** list, define when the parent process resumes processing:

Choices	Actions
<b>Resume processing after all subprocesses are complete</b>	Select <b>All</b> .

Choices	Actions
<p><b>Resume processing after any of the subprocesses is complete</b></p>	<p>Select <b>Any</b>.</p> <p>After any of the subprocesses are complete, processing of the remaining subprocesses stops and the system cancels open assignments.</p>
<p><b>Resume processing after a when condition returns a true value</b></p>	<ol style="list-style-type: none"> <li>Select <b>Some</b>.</li> <li>In the <b>Exit iteration</b> list, select <b>On when</b>.</li> <li>In the <b>When</b> field, enter a when condition that resumes processing after evaluating to true.</li> </ol>
<p><b>Resume processing after a specified number of subprocesses reach a certain status</b></p>	<ol style="list-style-type: none"> <li>Select <b>Some</b>.</li> <li>In the <b>Exit iteration</b> list, select <b>On count</b>.</li> <li>In the <b>Number of paths</b> field, enter an integer that determines how many subprocesses need to reach a certain status to resume processing.</li> </ol> <div data-bbox="938 1234 1461 1417" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 10px; margin: 10px 0;"> <p><b>Note:</b> Enter an integer that is lower or equal to the number of subprocesses in your process.</p> </div> <ol style="list-style-type: none"> <li>In the <b>Resulting with flow status</b> field, enter a status that subprocesses need to reach to resume processing.</li> </ol>

5. **Optional:** To provide a link to the Split Join shape step in the breadcrumb trail navigation, select the **Enable navigation link** check box, and then configure additional navigation options:



- To enable users to return to the step after the case moves forward, select the **Only allow navigating back to this step** check box.
  - To enable post-processing or validation when users navigate back from the Split Join shape step by using the breadcrumb trail, select the **Perform post-processing when navigating away from step** check box.
6. Open the subprocess configuration by expanding the **Specify a flow rule for this subprocess** section.
  7. In the **Name** field, enter a clear description of the subprocess.
  8. On the **Subprocess** tab, configure the context of the subprocess:

Choices	Actions
<p><b>The subprocess works on a current case</b></p>	<p>Select <b>On current page</b>.</p>
<p><b>The subprocess works on a specific case</b></p>	<ol style="list-style-type: none"> <li>Select <b>On specific work item</b>.</li> <li>In the <b>Work property</b> field, enter a property reference in the current work item that identifies the key of the subprocess work item.</li> <li>In the <b>Class</b> field, enter the class of the subprocess work item.</li> <li><b>Optional:</b> To specify a clipboard page that holds the subprocess work item, in the <b>Page name</b>, enter the clipboard page name.</li> </ol> <div data-bbox="938 1356 1468 1761" style="border: 1px solid #0070C0; padding: 10px; margin-top: 20px;"> <p><b>Note:</b></p> <p>If the case is open as a page on the clipboard, you can leave the <b>Work property</b> field blank.</p> <p>If the case does not already have a page and you do not provide a page name, your ap-</p> </div>

Choices	Actions
<p><b>The subprocess works on an embedded page</b></p>	<div data-bbox="850 212 1468 401" style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p>plication creates a page named pyNextObj, pyNextObj_1, and so on.</p> </div> <ol style="list-style-type: none"> <li>a. Select <b>On embedded page</b>.</li> <li>b. In the <b>Page property</b> field, enter a reference to the property of mode Page, Page List, or Page Group that holds the embedded page.</li> <li>c. In the <b>Class</b> field, enter the class of the embedded page.</li> </ol>

9. In the **Filter by flow** field, select the subprocess category:
  - To start a subprocess that consists of a series of actions to complete, select **Process flow**.
  - To start a subprocess that consists of a series of forms to complete, select **Screen flow**.
10. In the **Flow rule** field, press the Down arrow key, and then select the name of a subprocess to run.
11. **Optional:** Specify what information about the subprocess you want to include in the application documentation:
  - To link an application to the subprocess, in the **Application** field, enter the name of the application.
  - To provide a diagram of the subprocess in a documentation section that describes a specific case type, in the **Case/Supporting type** field, enter the name of the case type.
  - To provide implementation requirements of the subprocess, in the **Specification** field, enter the specification that holds the requirements.
12. **Optional:** To display a message in the audit trail of the subprocess, in the **Audit note** field, enter a rule that stores the audit note that you want to include.
13. Configure the second subprocess by repeating steps [6 \(on page 297\)](#) through [10 \(on page 298\)](#).
14. **Optional:** To add more subprocesses, click **Add a flow**, and then repeat steps [6 \(on page 297\)](#) through [10 \(on page 298\)](#).
15. Click **Submit**.
16. Click **Save**.

[Creating a stand-alone process \(on page 249\)](#)

[Changing the path of a process \(on page 280\)](#)

[Subprocesses \(on page 287\)](#)

[Flow shapes \(on page 251\)](#)



## Running multiple instances of the same subprocess with a Split for Each shape

Speed up case resolution by creating multiple instances of one subprocess that users can work on simultaneously and that later rejoin the parent process. Consequently, you provide tools for users to work in parallel and remove the risk of blocking users during case processing. In a sample scenario in which a case requires approval from multiple users, the Split for Each shape starts the same approval process for all users that meet specified conditions. For example, the approval process can begin for all users with managerial roles who can access the case. You can define how many approvals the case requires to move forward.

**Note:** Each subprocess runs asynchronously and in parallel. When a user works on an assignment in one subprocess, other users can work on an assignment in a parallel subprocess. The prerequisite for this process to work safely is to review the locking strategy, and consider switching to the multiple user locking strategy. However, concurrent access may result in a need to refresh the assignment due to the fact that another user can make changes at the same time.

1. Add the Split for Each shape to your case life cycle:
  - a. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - b. In the **Case life cycle** section, in the process in which you want to add the Split Join shape, click **Configure process**.
  - c. On the toolbar, click **Open process in Dev Studio**.
  - d. On the **Diagram** tab, click the **Flow shapes** icon, and then select **Advanced Shapes > Split for Each**.
  - e. Connect the **Split for Each** shape by dragging connector end points to connection points on different shapes in the process.
2. Open the dialog box by double-clicking the **Split for Each** shape.
3. **Optional:** To provide a unique name for the shape, in the **Split for Each** dialog box, in the **Split for Each** field, enter a new name.
4. In the **Join** list, define when the parent process resumes processing:
  - To resume processing after all subprocesses are complete, select **All**.
  - To resume processing after any subprocesses are complete, select **Any**.

After any of the subprocesses are complete, processing of the remaining subprocesses stops and the system cancels open assignments.


- To begin processing of individual pages for which a specified when condition evaluates to true, select **Iterate**.
  - To resume processing after a specified number of subprocesses is complete, select **Some**.
5. In the **Page property** field, enter the Page List or Page Group property that is the basis of the split.

The system autopopulates the **Class** field that stores the class with embedded pages of the property that you provide.

6. In the **Filter flow by** field, select the subprocess category:
- To start a subprocess that is a series of actions to complete, select **Process flow**.
  - To start a subprocess that is a series of forms to complete, select **Screen flow**.
7. In the **Flow name** field, enter the name of the subprocess that you want to run.
8. **Optional:** To control whether the flow runs for all or some pages in the page property, in the **When** field, enter a when condition that evaluates values in each page.  
At run time, the system evaluates this when condition rule once for each page of the property in the Page Property field and starts a subprocess for any page for which this rule evaluates to true.
9. If, in step 4 (on page 299), you select **Iterate** and your process resumes after a specified when condition evaluates to true, in the **Exit iteration when** field, enter a name of the when condition that resumes processing after evaluating to true.
10. If, in step 4 (on page 299), you select **Some** and your process resumes after a specified number of subprocesses is complete, configure the conditions to resume processing:

Choices	Actions
<b>Resume processing after a when condition returns a true value</b>	<p>a. In the <b>Exit iteration</b> list, select <b>On when</b>.</p> <p>b. In the <b>When</b> field, enter a when condition that resumes processing after evaluating to true.</p>
<b>Resume processing after a specified number of subprocesses reaches a certain status</b>	<p>a. In the <b>Exit iteration</b> list, select <b>On count</b>.</p> <p>b. In the <b>Number of paths</b> field, enter an integer that determines how many subprocesses need to reach a certain status to resume processing.</p>

**Note:** Enter an integer that is lower or equal to the number of embedded pages in the page

Choices	Actions
	<p data-bbox="941 226 1455 331">  property that you provided in step 5 <i>(on page 300)</i>.         </p> <p data-bbox="909 403 1455 529">           c. In the <b>Resulting with flow status</b> field, enter a status that subprocesses need to reach to resume processing.         </p>

11. **Optional:** To display a message in the audit trail of the subprocess, in the **Audit note** field, enter a rule that stores the audit note that you want to include.
12. **Optional:** To provide a link to the subprocess in the breadcrumb trail navigation, select the **Enable navigation link** check box, and then configure additional navigation options:
  - To enable users to get back to the subprocess after the case moves forward, select the **Only allow navigating back to this step** check box.
  - To enable post-processing or validation when users navigate back from the subprocess in the breadcrumb trail, select the **Perform post-processing when navigating away from step** check box.
  - If, in step 4 *(on page 299)*, you select **Iterate**, to allow users to return to the entry points in the subprocess, select the **Subprocess has navigation links** check box.
13. **Optional:** If in step 4 *(on page 299)* you selected **Iterate**, to define the order in which the system evaluates the pages, in the **Page group iteration settings** section, define exact values to evaluate at run time:
  - a. In the **Subscript order** field, enter text that your process compares with the subscript value for each page in the group.
  - b. **Optional:** To require an exact match, select the **Exact match** check box.
  - c. **Optional:** To ignore the pages in the group that do not match the subscript order that you provide, clear the **Process remaining changes** check box.  
Otherwise, your process evaluates the pages without a match after processing all conditions in the **Page group iteration settings** section.
14. Click **Submit**.
15. Click **Save**.

[Changing the path of a process \*\(on page 280\)\*](#)

[Flow shapes \*\(on page 251\)\*](#)



## About Flow Actions

A flow action controls how users interact with user forms to complete assignments. After selecting one flow action, users may fill in a section of the form to complete (perform) the assignment.

For relevant training materials, see the [Flow action processing](#) module on Pega Academy.

The following tabs are available on this form:

- Layout
- Validation
- Security
- HTML

### Where referenced

In a flow, developers can associate flow actions with connectors (arrows) and with assignment shapes. At run time, the flow actions associated with an assignment determine the choices available to users as they perform the assignment.

### Controls and Layouts

On the Layout tab, each flow action references a section that defines the runtime presentation of the flow action. You edit the section using controls and layouts.

### Short Descriptions

When you specify a description for the **Short Description** field, make the description meaningful in context to application users who must select one action from the list.

As a best practice, start the Short Description of a flow action with an action verb and describe a goal or sub-goal of the flow by using terms that users will recognize. For example, the following action labels convey the users' intent:

- Submit PO for approval
- Perform quality review
- Verify application

In contrast, offering users too many actions, or vaguely labeled actions, forces the user to learn and decide which actions are meaningful in that context.

### Action Section

On Perform user forms presented with the default action section ( `work-.pyActionArea` ), the form area created by a flow action can offer the most likely action as the default, while allowing the user to choose



other actions. The action with the highest likelihood appears as the default option on the left side of the section header. The section presents alternative actions in a click-action menu named **Other Actions**. The top of the menu displays connector flow actions, a horizontal line, and then local flow actions. Beneath the list of flow actions is a sub-menu called **Add Work**, which lets the user start a supporting process or add a child case — the choices depend upon your case management configuration.

The text in the action's Short Description field appears as user-visible text in the action selection list.

Other formats are supported.

#### Access

Use the Application Explorer to access the flow actions that apply to the work types in your application. Use the Records Explorer to list all the flow actions available to you.

After you complete initial development and testing, you can delegate selected rules to line managers or other non-developers. Consider which business changes might require rule updates and if delegation to a user or group of users is appropriate. For more details, see [Delegating a rule or data type \(on page 187\)](#).

#### Category


Flow action rules are part of the Process category. A flow action is an instance of the `Rule-Obj-FlowAction` rule type.

---

[Adding a validation rule to a flow action \(on page 187\)](#)

## Flow Actions - Completing the New or Save As form

A flow action has two key parts:

Field	Description
<b>Apply to</b>	<p>Select the name of a class, typically a class derived from the <code>Work-</code> base class.</p> <p>The list of available class names depends on the ruleset that you select. Each class can restrict applying rules to an explicit set of rulesets as specified on the <b>Advanced</b> tab of the class form.</p>
<p> <b>Note:</b> When work items in your application contain an embedded page or pages that can be any one of multiple <code>Data-</code> classes, choosing a <code>Data-</code> class for the <b>Apply to</b></p>	

Field	Description
	<p>key part of a flow action can be beneficial. See the Pega Community article <i>How to create flows that operate on embedded pages of a work item</i>.</p>
<b>Identifier</b>	Enter a name for this flow action. Begin the name with a letter and follow the rules for a Java identifier.

Additional creation options

Field	Description
<b>Template</b>	<p>To mark the new rule as JSR 168 compliant, select the name of a template rule to copy.</p> <p>Specify <code>PortletAction</code> to create a flow action that uses portlet-compliant HTML and can be displayed in a portlet window on a portal server through Service Portlet rules.</p> <p><b>Note:</b></p> <p>The rule type Service Portlet is deprecated. Use Pega Web Mashup instead.</p>

Rule resolution

When searching for instances of this rule type, the system uses full rule resolution which:

- Filters candidate rules based on a requestor's ruleset list of rulesets and versions
- Searches through ancestor classes in the class hierarchy for candidates when no matching rule is found in the starting class
- Finds circumstance-qualified rules that override base rules
- Finds time-qualified rules that override base rules

[About Flow Actions \(on page 302\)](#)

To create a Flow Action:

1. In the header of Dev Studio, click **Create > Process > Flow Action**.
2. On the **Create** form, enter values in the fields to define the context of the flow.





- a. In the **Label** field, enter text that describes the purpose of the circumstance definition.
  - b. **Optional:** To change the default identifier for the circumstance definition, click **Edit**, and then provide a unique value in the **Identifier** field.
  - c. **Optional:** Click **View additional configuration options** to include all other configuration options supported by this record type on the form. These options vary by record type and appear for only records that support [Quick Create \(on page 303\)](#) options.
  - d. Select a **Context** to specify where the record will reside in your application ruleset stack and how it may be reused in the class hierarchy.
  - e. In the **Apply to** field, press the Down Arrow key and select the class that defines the scope of the circumstance definition.
  - f. In the **Add to ruleset** field, select the name and version of a ruleset that stores the circumstance definition.
3. **Optional:** To override the default work item that your application associates with this development change, press the Down arrow key in the **Work item to associate** field, and then select a work item. For more information about your default work item, see [Setting your current work item \(on page \)](#).
  4. Click **Create and open**.

---

[About Flow Actions \(on page 302\)](#)

More about Flow Actions (on page )

## Configuring pre-processing for a flow action

Provide relevant data before users start working on an assignment by configuring pre-processing for a flow action. Flow actions can call a data transform, an activity, and a robotic automation, which gives you a possibility to meet your unique business requirements connected to data manipulation in a case.

For example, you can autopopulate fields on a form that a user needs to process to complete the assignment.

For relevant training materials, see the [Flow action processing](#) module on Pega Academy.

In pre-processing, the system first applies a data transform, runs an activity, and then runs an automation. Specify the elements of pre-processing that match your needs. For example, you can specify only a data transform, and you can omit providing robotic automation and an activity.



**Note:** If you assign this flow action to an assignment that is one of the possible paths in the flow, for example, a path that results from the Decision shape, you need to consider the likelihood value of the assignment. If this flow action has the highest likelihood value at run time, the form for this flow action appears as the default form when your application displays a Perform harness, and the



system also runs the data transform and the activity. As a result, users interact with the updated values on a form. For more information, see [Types of decision logic \(on page 282\)](#) and [Changing the decision logic in a process \(on page 283\)](#).



**Note:** You can configure pre-processing only for flow actions that do not support bulk-processing. Bulk processing cannot use a pre-processing activity, data transform, or a robotic automation.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. In the list of flow action instances, open the flow action that you want to edit.
4. On the **Action** tab, in the **Pre-processing** section, in the **Apply data transform** field, specify a data transform that the system applies to the current primary page at run time.
5. If the data transform includes parameters, in the **Parameters** section, enter values for the parameters.
6. In the **Run activity** field, specify an activity that the system runs before performing any additional processing for the flow action.

Consider the following factors when you use an activity for pre-processing:

- This activity runs only once, when a user selects the flow action for an assignment, even if the application displays the action area more than once. For example, when the data that the user submits for the first time fails validation. The activity also runs when you preview the flow action.
  - In the activity that you want to use, do not perform a Commit operation, and do not transfer the assignment.
7. If the activity has parameters, in the **Parameters** section, enter values for the parameters.
  8. If you use Robotic Desktop Automation (RDA), specify the pre-processing automation that you want to run on the user's desktop to retrieve data to display to users on a form in your application:
    - a. In the **Run robotic automation** field, enter an automation name.
    - b. In the **Description** field, enter text that describes what the automation does.



**Note:** Automations always run synchronously and flow processing pauses while the automation is in progress. After an automation completes processing, the system updates the clipboard, and renders the section that the flow action references.

9. Click **Save**.



## Related information

[Subprocesses \(on page 287\)](#)

[Calling one process from another process \(on page 292\)](#)

[Managing work across your team \(on page \)](#)

[Assigning tasks to users \(on page 84\)](#)

[Assigning users automatically at run time with business logic \(on page 85\)](#)

[Customizing the Get Next Work logic \(on page 317\)](#)

## Configuring post-processing for a flow action

Define the behavior of your application after users complete an assignment by providing a post-processing configuration for a flow action. With a post-processing configuration you can save time, automate your processes, and take advantage of additional options that improve manipulating data in a case. Because you can use a data transform, an activity, and a robotic automation, your application development process is flexible and can help you meet your individual requirements.

For example, you can use savable data pages to save information that the current case does not include. Consider a scenario in which a user who has an account in an online shop edits personal details when placing a shipping order. Although the modification of the contact information is external to the main case, a user can still save the relevant data.

For relevant training materials, see the [Flow action processing](#) module on Pega Academy.

To store modified information, you can use a top-level data page that you need to provide in the flow action. You can also use an embedded data page. However, in this scenario the new information persists only in the context of a current case and your application does not save the new data to any external system.

For post-processing, the system first invokes a robotic automation, saves the data pages, runs the data transform, and then runs an activity. Specify the elements of post-processing according to your needs. For example, you can specify only robotic automation and a data transform, and you can omit providing an activity.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. If you use the Robotic Desktop Automation (RDA), specify the post-processing automation to run on the user's desktop when you want to send data that users enter in a case form to all the applications that run on the user's desktop:
  - a. In the **Run robotic automation** field, enter an automation name.
  - b. In the **Description** field, enter text that describes the purpose of the automation.



**Note:** Automations always run synchronously and flow processing pauses while the automation is in progress. After the automation stops processing, the system updates the clipboard, and then the data transform runs.

4. **Optional:** To save a data page after the flow action completes, in the **Savable data pages** subsection, list the data pages that you want to save:

Choices	Actions
<p><b>Use a data page</b></p>	<ol style="list-style-type: none"> <li>Click <b>Add data page</b>.</li> <li><b>Optional:</b> To save the data page only under certain circumstances, in the <b>When</b> field, enter a When condition rule. At run time, if the When rule evaluates to true, the application saves the data page. If you leave this field blank, the application always saves the data page.</li> <li>In the <b>Data page</b> field, specify a data page to save.</li> </ol>
<p><b>Use a data relationship by providing an autopopulated property</b></p>	<ol style="list-style-type: none"> <li>Click <b>Add data page</b>.</li> <li><b>Optional:</b> To save the data page only under certain circumstances, in the <b>When</b> field, enter a When condition rule. At run time, if the When rule evaluates to true, the application saves the data page. If you leave this field empty, the application always saves the data page.</li> <li>Select the <b>Use associated property</b> check box.</li> <li>In the <b>Autopopulate property name</b> field, specify an autopopulated property.</li> </ol>

5. **Optional:** To add more data pages, click **Add data page**, and then repeat step 4 *(on page 308)*.
6. **Optional:** To monetize efforts connected with performing the flow action, in the **Apply cost** field, specify the flow action cost:
  - To specify the cost manually, enter an expression or a number in units of any type, such as currency.
  - To calculate the value automatically, enter a property reference that identifies a numeric property that is used in your computation.

In a single application, use only numbers or property references.

7. In the **Apply data transform** field, specify a data transform that the system applies to the current primary page at run time.
8. In the **Run activity** field, enter the activity that the system runs after processing the flow action is complete.

Consider the following factors when you use an activity for post-processing:

- Do not include a Commit method in the activity.
  - If you use an activity that depends on parameter values that a user enters after selecting this flow action, enter the activity in the post-processing configuration rather than placing the activity in the flow.
  - You can use the activity to perform validation processing and add messages to the clipboard if validation fails. Validation by applying an activity is more suitable for complex validation scenarios. The activity can also perform additional processing that occurs after the user successfully submits the flow action.
9. If the activity includes parameters, in the **Parameters** section, enter values for the parameters.
  10. Click **Save**.

### Related information

[Subprocesses \(on page 287\)](#)

[Calling one process from another process \(on page 292\)](#)

[Managing work across your team \(on page \)](#)

[Assigning tasks to users \(on page 84\)](#)

[Assigning users automatically at run time with business logic \(on page 85\)](#)

[Customizing the Get Next Work logic \(on page 317\)](#)

## Configuring back-to-back processing for a flow action

Accelerate case resolution by allowing users to perform more assignments in a case after processing of the current task is complete. When an application routes multiple assignments to a single user, performing the assignments in sequence increases efficiency and promotes uninterrupted workflow. For example, a flow might require the editing and sending of three items of outgoing correspondence. In the flow, a Split



Join shape might cause Pega Platform to create three correspondence editing assignments, all in a single worklist. After a user edits the first correspondence item and completes one of the three assignments, the user can immediately continue processing the remaining two assignments. Back-to-back processing is optional but considerably increases the productivity and efficiency of your system.

Configuring back-to-back processing is a part of activity post-processing. Post-processing defines which actions users can take after they complete a task in a case.

After you configure back-to-back processing, the system finds and displays only assignments that the current user is qualified to perform and for which the value of the *pyActionTime* property from the *Assign-* class is past.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. In the **Post-processing** section, in the **Back-to-back processing configuration** subsection, select the **Look for an assignment to perform** check box.

At run time, the system searches for another assignment for the same case and same flow execution on the user's worklist

4. **Optional:** To ensure a continuous workflow, specify how the system behaves if the user's worklist does not contain any relevant assignments:
  - To search for open assignments in the user's worklist from other flow executions that are active for the same case, select **If not found, look for assignments in other flows on this case**.
  - To search for open assignments from the parent case of the current case, select **If not found, look for assignments in flows on the parent case**.
  - To search for assignments in work queues, select **For each also consider assignments in work queues**

The search scope expands to examine the assignments in the work queues of the user's team and assignments on the user's worklist.

Optionally, your application can use additional information for the detailed search criteria that the system uses when you select this check box. The standard decision tree *Assign-Workbasket.PerformCriteria* defines these criteria. For more information about decision trees, see *Decision trees (on page )*.

5. In the **If an assignment is not being performed** list, select content that the system displays to users after all of the assignments are complete:

- To display a refreshed harness form for the current case, select **Show Harness**, and then in the text field, enter the relevant harness.

For example, you can use a standard post-assignment `Confirm` harness that displays the results of processing or computations based on user input.



**Note:** Avoid specifying a harness of the `Perform` type that displays the case in edit mode.

- To configure the system to search for an assignment by using the Get Next Work logic, select the **Get Next Work**.

The system uses information about the user's skillset and urgency of available assignments to calculate the next assignment for the user to process. For more information about the Get Next Work algorithm, see [Customizing the Get Next Work logic \(on page 317\)](#) and [Defining work routing settings for an operator \(on page 317\)](#).

- To configure the system to select the next assignment in the workpool that is assigned to the current user, select **Get Next Work in Current WorkPool**.
- To close the user form and return to the previous work area, such as the Process Work home view, select **Close the Work Object**.



**Note:** This setting applies only to **Submit** and **Next/Finish** action buttons. For more information about action buttons, see [Configuring action buttons for a flow action \(on page 311\)](#).

6. Click **Save**.

## Configuring action buttons for a flow action

Specify how users can interact with an assignment in a case by defining buttons on a flow action. To meet your unique business needs, assign your own custom labels to the buttons so that users know exactly what action takes place after clicking the button. Consider a scenario in which a user places an order in an online shop. After the user provides their shipping details, a review form with the order details opens. Custom action buttons help the user return to the editable form to change their details. The labels also communicate that in the next step in the flow, the user chooses a payment method.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. In the list of flow action instances, open the flow action that you want to edit.
4. On the **Action** tab, in the **Action buttons** section, define buttons with which users can interact:

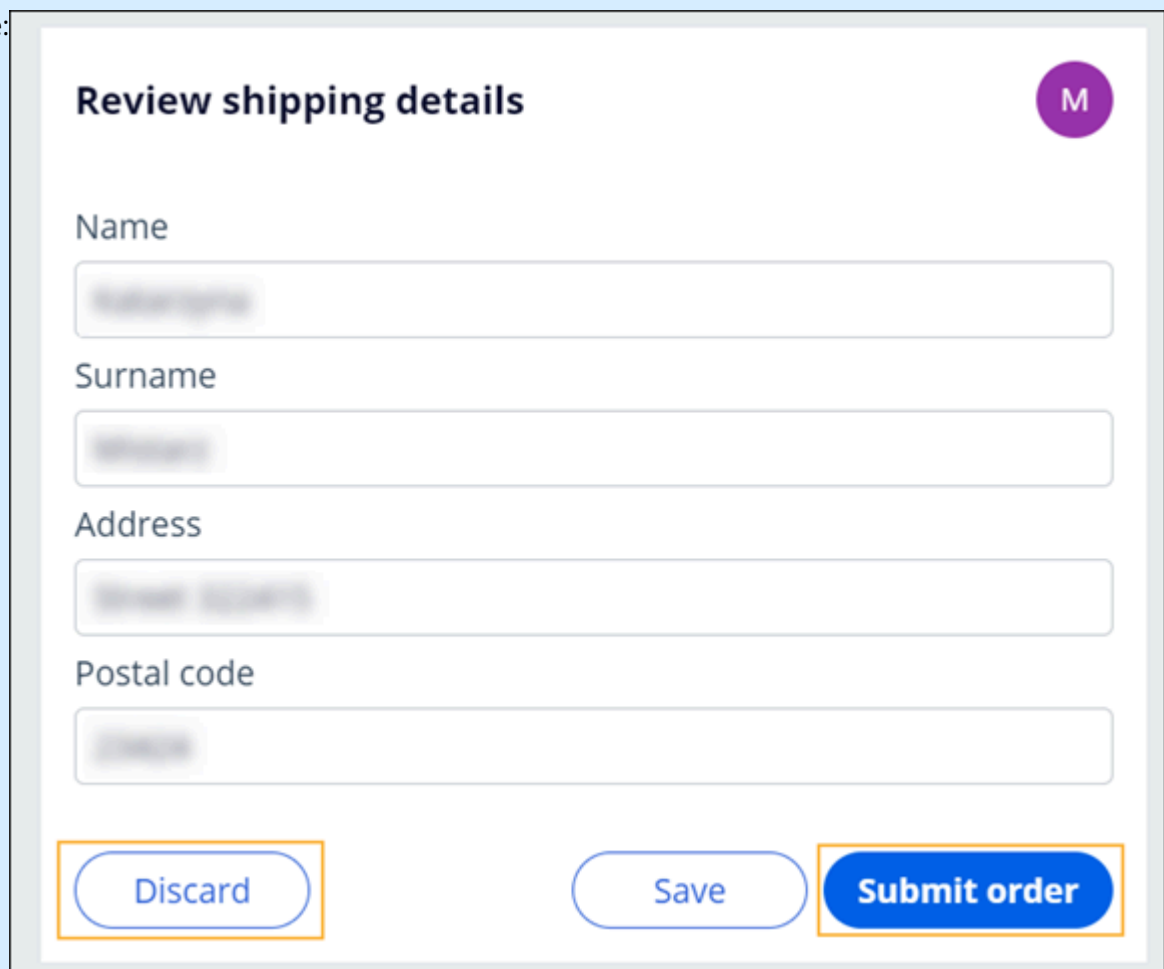
Choices	Actions
<p><b>Hide default action buttons</b></p>	<p>Select the <b>Hide the default action section buttons</b> check box.</p> <p>If you select this check box, ensure that you provide custom navigation, such as buttons, drop-down menus, and links, in the body of the action section. For example, use the <i>px-Button</i> control and a Cancel action instead of the <b>Cancel</b> action button. If you do not provide an alternative navigation method, users cannot proceed with a case. For more information, see <i>Configuring a Button control (on page )</i>.</p>
<p><b>Customize the text on the default navigation buttons</b></p>	<ol style="list-style-type: none"> <li>a. Select the <b>Customize the action section buttons labels</b> check box.</li> <li>b. In the field of a relevant action button, enter your custom text. You can customize the following action button labels: <ul style="list-style-type: none"> <li>• <b>Submit button label</b> that submits the current work item. This button typically only appears in the last step of a flow action.</li> <li>• <b>Previous button label</b> that displays the previous step in the flow action, if one exists. This button only appears after the user moves past the first step.</li> <li>• <b>Cancel button label</b> that cancels the process and closes the flow action without saving any changes.</li> </ul> </li> </ol>



Choices	Actions
	<ul style="list-style-type: none"> <li>• <b>Next button label</b> that displays the next step in the flow action, if one exists. This button does not appear in the last step of the flow action.</li> </ul>

5. Click **Save**.

At run time, users interact with buttons that have the customized, descriptive labels, as in the following figure:



The screenshot shows a form titled "Review shipping details" with a purple circular icon containing the letter "M" in the top right corner. The form contains four input fields: "Name", "Surname", "Address", and "Postal code". At the bottom of the form, there are three buttons: "Discard" (light blue with a blue border), "Save" (light blue with a blue border), and "Submit order" (dark blue with a white border). The "Discard" and "Submit order" buttons are highlighted with orange rectangular boxes.

Custom buttons on an order form

Hybrid mode

Working with Dev Studio forms (on page )

Subprocesses (on page 287)

Managing work across your team (on page )

[Assigning tasks to users \(on page 84\)](#)

[Assigning users automatically at run time with business logic \(on page 85\)](#)

## Configuring indicator for a flow action

To design cases that precisely meet your business requirements, define the results that user interactions with a case have by placing a flow action at a relevant place in a flow. With flow actions, users can update information in an assignment, resolve work, or both. Consider a scenario in which a customer service representative (CSR) submits a form with a loan request from a customer. You can decide whether the CSR only updates information in a case or whether the CSR can also resolve the assignment and move the case to the next task. An indicator on an assignment or a connector between assignments determines where in a flow that flow action happens.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. In the list of flow action instances, open the flow action that you want to edit.
4. On the **Action** tab, in the **Indicator** section, in the **Used as** list, specify how you intend to use the flow action:

- To update the assignment and the current case, select **Local Action**.

By performing local actions, users can update, but not resolve, an assignment. The local action retains the flow at the same assignment shape, so that the current user, or another user in some cases, can choose another flow action. Users can invoke local flow actions from a modal window or from a flow action form.

- To update the current case, complete the assignment, and advance the case, select **Connector Action**.

Typically, users invoke connector flow actions from a flow action form. Local flow actions are alternatives to autogenerated controls.

- To use the flow action as both a local action and a connector action, select **Local and Connector**.

5. **Optional:** To exclude the flow action from the list of flow actions that are available for bulk processing, select **Disqualify this action from bulk processing**.

You can perform bulk processing of flow actions in the following situations:

- The flow action does not have circumstances.
- For assignments in various cases, the `Applies To` class of the flow action is a class that is an ancestor to all the cases. For more information about classes, see [Understanding class hierarchy and inheritance \(on page 84\)](#).

By default, some standard flow actions do not support bulk processing because the process can involve only one case at a time. To change the default setting, copy the flow action to your ruleset, and then make the change. For more information, see [Copying a rule or data instance \(on page 477\)](#).

6. Click **Save**.

---

[Running flow actions in bulk \(on page 477\)](#)

## Managing case types

Reflect your individual business needs and requirements by modifying your case types. You can rename, import, or delete a case type. For example, if your business requirements change and one of your case types is no longer relevant, you can delete that case type.

---

[Case types \(on page 28\)](#)

## Importing a case type

To reduce the complexity of your application, share templates for business processes between your applications by importing case types. For example, when you have multiple applications to deal with different kinds of loan requests, you can share the template for the loan request review process.

1. Define a build-on application from which you want to import the case types:
  - a. In the navigation pane of Dev Studio, click **App**.
  - b. Under the name of your application work pool, click **SysAdmin > Class**.
  - c. In the **Class inheritance** section, in the **Parent class (Directed)** field, enter the work pool of your built-on application.
  - d. Click **Save**.
2. In the navigation pane of Dev Studio, click **Case types**.
3. In the navigation pane of App Studio, click **Case types**.
4. In the **Case types** panel, click **Options > Import case types**.
5. Review the list of top-level case types that you add to your application.

Case types with the same name as an existing case type in your application do not appear in this list.

6. **Optional:** To exclude a case type from the import process, clear the check box next to the name of that case type.
7. Click **Submit**.



The case type and its child case types now exist in your application. Supporting rules, such as the starting flow for a case type, continue to reside in the built-on application. You can leave them there, or create specialized versions in your application.

[Case types \(on page 28\)](#)

[Extension points for case-type creation \(on page 316\)](#)

## Extension points for case-type creation

You can use extension points to customize the processing that occurs when you add or import case types to your application. For example, you can copy records from built-on applications to reuse resources.

The following extension points are supported by the New Application wizard and the Case Type Explorer:

- `@baseclass.pyIsFrameworkAssetGenerationForCasesEnabled`
- `@baseclass.pyPrepareFrameworkAssetsForGenerationSetup`
- `Rule-Obj-CaseType.pyPrepareFrameworkAssetsForGeneration`
- `@baseclass.pyPrepareFrameworkAssetsForGenerationPost`

Do not use the **Obj-Save** or **Commit** methods in your overridden versions because it can lead to unexpected results. All records that are created by your extension points are saved to the database by an activity that is higher in the call stack.

[Importing a case type \(on page 315\)](#)

[Case types \(on page 28\)](#)

## Renaming case types

Reflect changes to your business requirements by updating the name of a case type. For example, to add a car insurance claim to your homeowners insurance claim, you can rename your case type to `Car` and `homeowners insurance claim`.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case type** field, enter a new name for your case type.
4. In the **Edit case type** field, click the case type name.



5. In the **Short Description** field, enter a new name for your case type, and then click **Submit**.
6. Click **Save**.

[Case types \(on page 28\)](#)

## Deleting case types

Delete a case type and all associated assets in your application to remove the records that you do not need anymore, for example, when the case type no longer meets your business requirements. You can also delete only the associated case instances, such as sample cases that you do not need after you finish development. Removing irrelevant assets saves you time and money through the efficient management of your database space.

1. In the navigation pane of App Studio, click **Case types**.
2. In the row of the case type that you want to delete, choose the scope of your change:
  - To delete the case type and all associated assets, click **More options > Delete case type**.  
  
The assets that are associated to a case type include a data model, life cycle, and the UI elements.
  - To delete all case instances, attachments, Pulse conversations, and assignments for the case type, click **More options > Delete cases**.
3. Click **OK**.

[Case types \(on page 28\)](#)

## Customizing the Get Next Work logic

You can customize Get Next Work processing to meet the needs of your application and your business operations without creating new activities. If you create or override standard rules, save the rules in a ruleset and version that is available to all appropriate users.

1. Ensure that the assignment urgency ( `pxUrgencyAssign` ) is set to a meaningful value, as this is a basis for sorting and prioritization. By default, a standard Declare Expression rule computes this value based on work item urgency and other criteria. For example, you can override the urgency value to base the assignment urgency on dollar amounts, customer categories, specific assignments that are often on the critical path, work item age, work types, and so on.
2. If a user can access assignments from multiple applications and work types, verify that the computation of assignment urgency is appropriate across the applications and work types. For example, in some business settings, it may be appropriate to complete every assignment in the CustomerOrder application before any assignments in the ResearchingVeryOldRocks application.



3. Confirm that the Get from work queues first check box on the Work tab of the Operator ID form is selected or cleared for affected operators.
4. If an assignment has skills that are designated as required, ensure that the values for those skills in the Operator ID instances are correct for the operators who have those skills. Confirm that the skills are necessary to complete the assignment.



**Note:** In flows, you can use the standard router activity `ToSkilledWorkBasket` to send an assignment with required or desired skills to a work queue.

5. If work queues are searched, review the work queues for each operator to ensure that the order in which they are listed in the Work tab of the Operator ID form is correct.
6. Override the Application Settings rule `GetNextWork_WorkBasketUrgencyThreshold` to identify a minimum cutoff value for assignment urgency. Overriding this value can speed the entire operation by changing the order in which candidates are examined. First, it causes searches of only those assignments in a work queue that equal or exceed an urgency threshold; if none is found that meet all the criteria, the next work queue is searched. Only after all the work queues are searched with no qualifying assignment found are the work queues revisited to look for assignments with lower urgency. To simplify testing, you can supply different cutoffs for different production levels.
7. Copy and override the list value rules to change the 500 limit on assignments examined or to add additional selection criteria.
8. During testing and in production, note the situations in which the algorithm that you have implemented do not return any assignments. Also note where and how the human operator looks next for assignments.

### Changing case statuses

Communicate to stakeholders how a business process moves towards resolution, by changing the case status in the life cycle. For example, you can assure the stakeholders that the case advances within the time frame.

You can configure the status change when the case reaches a certain stage or step.

For relevant training materials, see the [Case status](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, define when the case changes the status:

Choices	Actions
<p><b>Change the status when the case reaches a specific stage</b></p>	<ol style="list-style-type: none"> <li>Click the header of the stage that you want to configure.</li> <li>In the <b>Stage</b> properties pane, on the <b>General</b> tab, in the <b>Set case status on stage entry</b> field, press the Down arrow key, and then select a value that you want to apply.</li> </ol>
<p><b>Change the status when the case reaches a specific step</b></p>	<ol style="list-style-type: none"> <li>Click the step that you want to configure.</li> <li>In the <b>Step</b> properties pane, on the <b>General</b> tab, in the <b>Set case status</b> field, press the Down arrow key, and then select a value that you want to apply.</li> </ol>

5. Click **Save**.

At run time, the case changes the status when the case reaches a point that you define in the case life cycle.

### Related information

[Defining conditions for skipping a stage \(on page 63\)](#)

## Case status values

The value of a case status is a primary source of information for users about the current state of a business process. For example, by analyzing a case status, users can learn whether a case is already in progress or awaits an action, and then make informed decisions about actions to take. A detailed case status conveys the essential information about a business process, and as a result, promotes a unified understanding of a case between the stakeholders that the case involves. For example, a `Pending-Approval` status informs users that a case awaits approval.

Changes to a case status convey important information about the progress of the case towards completion. In addition, the system uses the case status to control automatic case instantiation.

The following table describes typical values of a case status:



Case status	Description
New	Indicates that the case is not yet in progress. The status does not trigger any case processing.
Open	Indicates that the case is active and that users process the case. The status does not trigger any case processing. You can extend the <code>Open</code> status to provide more details. For example, you can create an <code>Open-Sales</code> status to inform users that a sales team currently processes the case.
Pending	Indicates that the case awaits an action or an event. The status does not trigger any case processing. You can extend the <code>Pending</code> status to specialize the meaning. For example, in a hiring process, you can create a custom <code>Pending-ReferenceCheck</code> status to inform users that a case can only move forward after a designated user checks the references of a job candidate.
Resolved	Indicates that all processing of the case is complete and the case does not include any open assignments. The status triggers particular actions in a case processing that typically happen as the last step in the case. You can also extend the <code>Resolved</code> status. For example, in a loan request case, you can create a custom <code>Resolved-LoanGranted</code> status to communicate that the loan request is approved, or a custom <code>Resolved-Rejected</code> to inform users about the rejection of a case.

For consistency and better understanding of case statuses, as a best practice, extend existing statuses instead of creating new values. In Dev Studio, the `Work-.pyStatusWork` property corresponds with case status.

To meet your unique business needs, you can create new or modify existing statuses. For more information, see [Defining a custom case status \(on page 321\)](#).

[Standard rules and extension points for case status \(on page 321\)](#)

[Assigning tasks to users \(on page 84\)](#)

[Assigning users automatically at run time with business logic \(on page 85\)](#)

[Completing work on time \(on page 324\)](#)

[Creating and managing cases \(on page 463\)](#)

[#unique\\_270 \(on page \)](#)

[Reopening a resolved case \(on page 474\)](#)





## Defining a custom case status

Define a custom case status to more precisely communicate the progress that a case makes toward resolution.

1. In the header of Dev Studio, click **Configure > Case Management > Processes > Status Values**.
2. Review the list of case statuses, and ensure that a case status that meets your business needs does not already exist.
3. Create a field value that defines your custom status by doing the following:
  - a. In the header of Dev Studio, click **Create > Data Model > Field Value**.
  - b. In the **Label** field, enter the text for the custom status value.

As a best practice, use one of the following prefixes to support standard reports and maintain continuity in between your custom status and standard status values: `New-`, `Open-`, `Pending-`, or `Resolved-`. Use only alphanumeric symbols in case statuses.

- c. In the **Field Name** field, enter **pyStatusWork**.  
At run time, your application maps a value of the pyStatusWork property with a case status.
- d. In the **Context** section, select an application layer, applies to class, and ruleset based on how you plan to reuse the status value in your application.



**Tip:** If your application supports localization, select a language-specific ruleset for the field value.

- e. Click **Create and open**.
- f. **Optional:** To provide a translated value for the custom status value, enter text in the **To** field of the Field Value form.
- g. Click **Save**.

---

[Collecting information from a user \(on page 83\)](#)

[Changing case statuses \(on page 318\)](#)

## Standard rules and extension points for case status

You can use standard rules and extension points to programmatically update, retrieve, or analyze the status of a case.



You can reference or override the following standard rules:

- `Work-.UpdateStatus` — An activity that changes the status of a case based on the value that you provide.
- `Work-.WorkBasket` — An activity that puts an assignment in a work queue and then calls `Work-.UpdateStatus` to change the case status.
- `Work-.Worklist` — An activity that puts an assignment on a worklist and then calls `Work-.UpdateStatus` to change the case status.
- `Work-.pyStatusWork` — A property that stores the current status of a case.

Do not set the value of this property because the status that you provide might not be detected by case-dependency logic. Use an activity, flow shape, or case life cycle to set the case status instead.

- `Work-.pyElapsedStatusNew` — A property that stores the cumulative time, measured in seconds, that a case has spent in any status with the `New` prefix.
- `Work-.pyElapsedStatusOpen` — A property that stores the cumulative time, measured in seconds, that a case has spent in any status with the `Open` prefix.
- `Work-.pyElapsedStatusPending` — A property that stores the cumulative time, measured in seconds, that a case has spent in any status with the `Pending` prefix.
- `Work-.Status-Resolved` — A ticket that is raised when a case is resolved to communicate the status change to other business processes.

---

[Changing case statuses \(on page 318\)](#)

[Enabling automatic conditional creation of child cases \(on page 224\)](#)

[Defining a custom case status \(on page 321\)](#)

## Changing a stage with open assignments

Create a flexible application that meets your business needs even in untypical scenarios by configuring a case to change stage without resolving open assignments in a current stage. As a result, users can complete a specific assignment at any time during a case life cycle. Consider the example of a recruitment process. The Review stage might include running a background check that an HR worker might need to additionally perform later on in a case, after the reviews of the documents from the candidate and the score from the job interview.

By default, when a case moves to the next or another stage, an application resolves open assignments in the current stage. For example, in a recruitment case, when your case moves from the `Review` to `Offer`

stage, the application resolves open assignments in the **Review** stage. If you want to keep assignments open, configure stage transition by using the **Utility** shape, instead of typical shapes, to move a case further in a case life cycle.

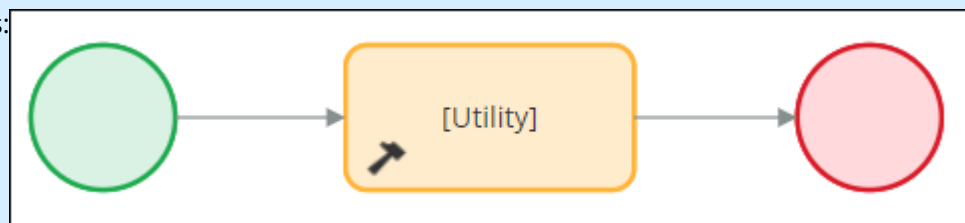
As a best practice, create a process that stores the **Utility** shape, and then add the process in an appropriate stage in your case life cycle.

1. Create a process with the **Utility** shape:
  - a. In the header of Dev Studio, click **Create > Process > Flow**.

**Note:** A flow is a legacy name for a process.

- b. In the **Label** field, briefly describe the purpose of the process.
  - c. In the **Context** section, select an application layer to store the process.
  - d. **Optional:** To change the default identifier for the process, click **Edit**, and then provide a unique value in the **Identifier** field.
  - e. In the **Apply to** field, click the Down arrow key, and then select a class that stores the process.
  - f. In the **Add to ruleset** list, select a ruleset and a ruleset version to store the process.
  - g. Click **Create and open**.
2. On the **Diagram** tab, remove the default **Assignment** shape by selecting the shape and clicking **Delete**.
3. On the toolbar, click **Flow Shapes**, and then click **Utility**.
4. Connect the **Utility** shape by dragging the connector end points to the connection points on the **Start** and **End** shapes.


The following figure shows a process with a **Utility** shape connected to the **Start** and **End** shapes:



Utility shape in a process

5. Open the dialog box by double-clicking the **Utility** shape.
6. **Optional:** To provide a unique name for the shape, in the **Utility properties** dialog box, in the **Utility** field, enter a new name.

7. In the **Automation details** section, in the **Select type of rule** list, select **Activity**.
8. In the **Rule** field, enter `pxChangeStage`.
9. In the **Parameters** section, define the stage that the case enters from the Utility shape:
  - To move the case to the next stage, select the **ChangeToNextStage** check box.
  - To move the case to another stage in the case life cycle, in the **ChangeToStage** field, enter a stage ID or stage name.

**Tip:** You can configure the Utility shape to move the case to the next stage by selecting the  **ChangeToNextStage** check box, and then providing another stage that the case enters when the **ChangeToNextStage** check box returns a false or invalid value.

10. **Optional:** To provide additional information for auditing reasons, in the **AuditNote** field, enter the audit note to add to the case history when the case moves to the Utility shape.
11. Leave the assignments open after the case moves to another stage by clearing the **CleanUpProcesses** check box.
12. Define properties to capture the run-time output details:
  - a. In the **CurrentStageLabel** field, enter a label for the current stage to display at run time.
  - b. In the **CurrentStage** field, enter the ID for the current stage.
  - c. In the **AutomationErrors** field, enter a page of the *Code-Automation-Errors* class that stores the details if the *pxChangeStage* activity fails during processing.
13. Click **Submit**.

[Processes in a case life cycle \(on page 66\)](#)

[Adding a parallel process to a stage \(on page 69\)](#)

[Empowering knowledge workers \(on page 348\)](#)

[Building case types \(on page 20\)](#)

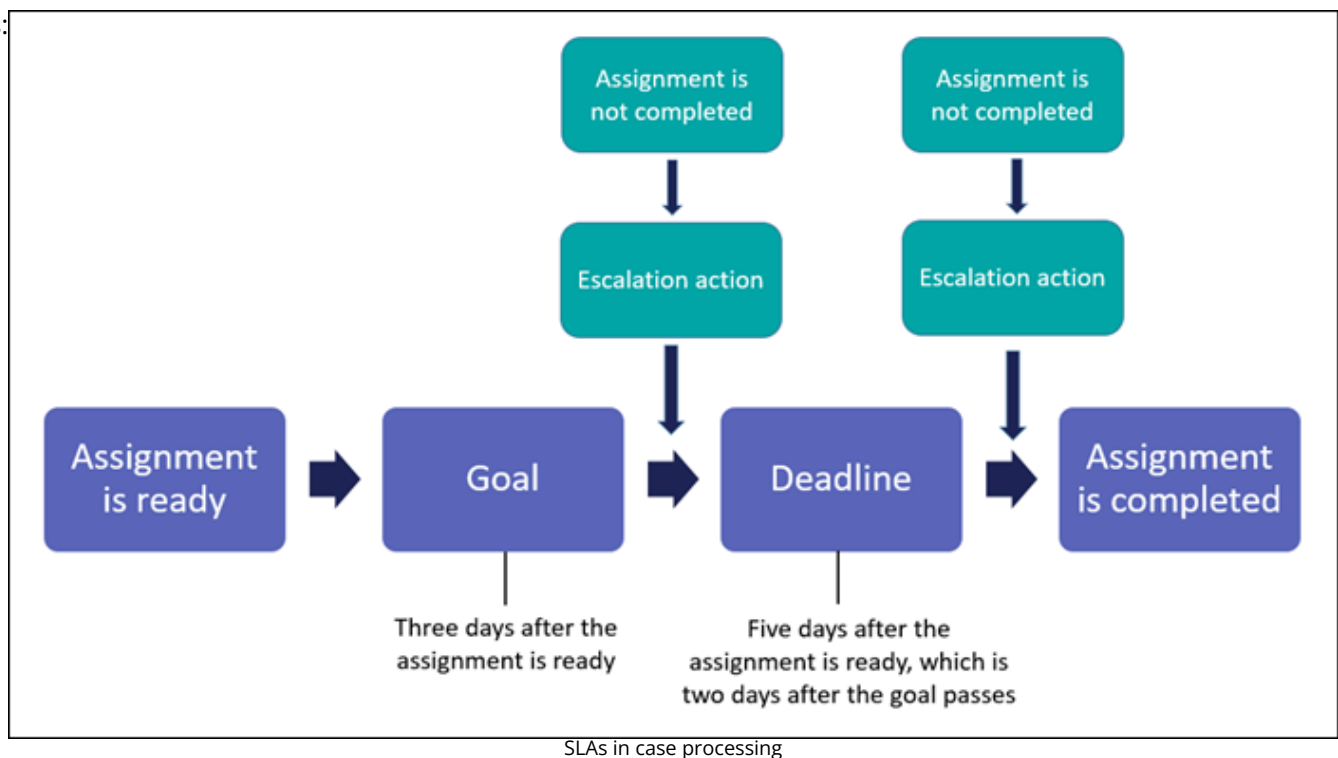
## Completing work on time

Enforce the service-level agreements (SLAs) that you make with customers and stakeholders by creating goals and deadlines for case workers. By implementing SLAs, you make a commitment to your customers and agree on a timeline for delivery of your projects.

A service-level agreement (SLA) defines a goal and deadline, which are intervals of time that you can apply to a case or elements in the life cycle of a case. By using SLAs, you can standardize the way that case workers resolve cases in your application.

For relevant training materials, see the [Completing work on time](#) and [Extending service-level agreement configurations](#) modules on Pega Academy.

When you define a goal for an assignment, you define the suggested time required to resolve work. When you define a deadline, you define the ultimate time in which to resolve work. You can also define escalation actions that your application performs when the goal and deadline expire. In a sample scenario, the goal for an assignment is three days and the deadline is five days. An application starts calculating the goal and deadline after the assignment is ready for processing. If a case worker misses to complete the assignment within three days, an escalation action occurs and the worker receives a notification about an elapsed goal. If the case worker fails to complete work within the next two days, another escalation action takes place and the manager of the case worker receives a notification about a passed deadline. Finally, the worker completes the assignment. The following figure shows the process:



You can apply SLAs to the following elements of case processing:

### Case

Starts when a user starts or reopens a case, and stops when the case reaches resolution.

### Stage

Starts when a case enters the stage, and ends when the case leaves the stage. You can apply SLAs to primary and alternate stages, and to optional stages.

### Process in a stage

Starts when a process starts, and ends when the last step in the process finishes. You can apply SLAs to processes that are part of a case life cycle and to optional processes.

## Assignment

Starts when a user or an automation creates or routes an assignment to a work queue or worklist so that the assignment is ready for processing, and ends when the assignment is completed or stops due to an error condition.

The time at which a user opens the form for an assignment has no effect on the service-level agreement.

## Approval step

Starts after a user receives the approval step that is ready for processing, and ends when the step is completed.

To help workers prioritize work, each assignment includes urgency, which is a numeric value that indicates how important the assignment is. You can decide how the urgency increases after a goal or deadline elapses to bring attention of workers to the tasks that are top priority and need immediate processing.

---

[Automating work by creating case types \(on page 27\)](#)

## Urgency

Urgency is a numeric value, ranging from 0 to 100, that worklists and cases display to indicate the importance of every assignment. Use urgency to bring visibility to unresolved assignments and cases in your application that need immediate action from case workers. As a result, you ensure that cases reach resolution within the timelines that you define with your stakeholders.

The higher the urgency, the more important it is to address the unresolved item. The initial urgency value is 10. You can define how elapsed goals and deadlines affect the urgency of a case, a stage, a process, an assignment, and an approval step, directly in Case Designer. The

following figure shows the configuration for increasing urgency after a goal for an assignment

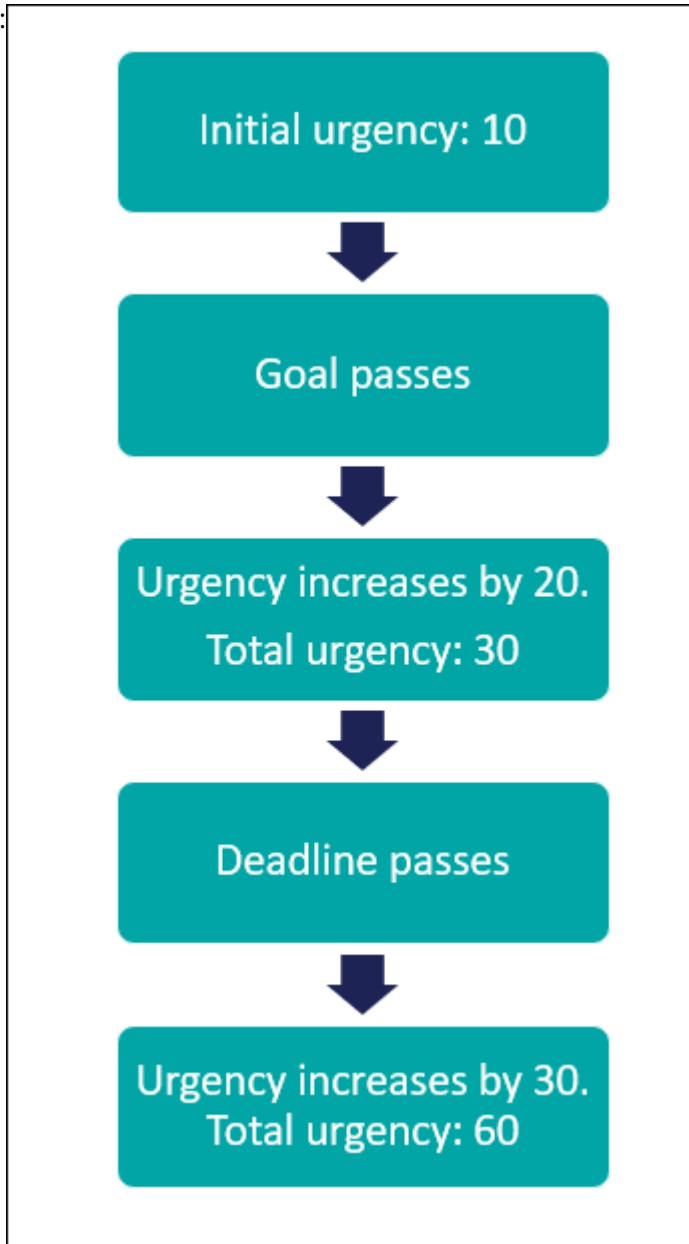
passes:

The screenshot displays a configuration window for a step named 'Review'. The left pane shows a list of tasks: 'Review documents', 'Review income data', 'Review banking history' (highlighted with an orange box), and 'Send email'. Below these are '+ STEP' and '+ FORM STEP' buttons. The right pane is titled 'Step' and has two tabs: 'General' and 'Goal & deadline'. The 'Goal & deadline' tab is active, showing instructions to define completion times. It includes a 'Use Service-level agreement (SLA)' dropdown set to 'Custom SLA'. Under the 'Goal' section, there are input fields for 'Days' (value: 3) and 'HH:MM:SS' (value: 00:00:00). Below these is an 'Increase urgency by' field with the value 20, which is also highlighted with an orange box. An orange line connects the 'Review banking history' task in the left pane to the 'Increase urgency by' field in the right pane.

Increasing urgency

Consider a scenario of a loan request case that includes an assignment to review the banking history of a customer who applies for a loan. The goal for the assignment is three days, and if the customer service representative (CSR) fails to meet the goal, the assignment urgency increases by 20. With an initial urgency of 10, the total urgency is then 30. If the CSR misses the deadline for the assignment, which is two days in this sample scenario, the

urgency rises again, by 30, and totals at 60. The following figure presents how the urgency rises:



Urgency changes throughout assignment processing

Additionally, in Dev Studio, you can adjust initial urgency when you create a service-level agreement rule. You can then apply the rule to a case, a stage, a process, or a step. You can use this option to prioritize certain work in your application, for example, tasks that concern VIP customers or that are critical to your organization. You can also use the following rules to change urgency:

#### **Work-.UrgencyUpdate**

A flow action that you can call during case processing. An application adds the integer that you provide to the current urgency of the case.

#### **Work-.pyUrgencyWorkAdjust**



A property that you can set to increase or decrease the current urgency of a case. The *Work-.pxUrgencyWork* declare expression evaluates the value that you provide.

### **Assign-.pyUrgencyAssignAdjust**

A property that you can set to increase or decrease the current urgency of an assignment. The *Assign-.pxUrgencyAssign* declare expression evaluates the value that you provide.

At run time, the system updates urgency each time that the value of a standard property changes or a user saves a case.

[Completing work on time \(on page 324\)](#)

[Setting service-level agreements \(SLAs\) for case resolution \(on page 329\)](#)

[Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#)

[Creating a service-level agreement \(SLA\) rule \(on page 332\)](#)

[Managing work across your team \(on page \)](#)

## **Setting service-level agreements (SLAs) for case resolution**

Encourage customer service representatives (CSRs) to resolve cases on time and enforce your service-level agreements (SLAs) by setting goals and deadlines for a case type. As a result, you save time and resources because you avoid missing deadlines. SLA measurement starts when the case starts and ends when the case finishes.

For example, if a CSR of an insurance claim request does not resolve the request within a week, the urgency of this case rises and the manager of the CSR receives a notification.

For relevant training materials, see the [Completing work on time](#) module on Pega Academy.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Settings** tab, click **Goal & deadline**.
4. In the **Use Service-level agreement (SLA)** list, define the SLA criteria:
  - To define custom criteria, select **Custom SLA**, and then go to step [5 \(on page 329\)](#).
  - To reuse criteria from an existing SLA, select **Existing SLA**. In the auto-complete field, enter the SLA name, and then go to step [10 \(on page 330\)](#).
  - To leave a case without an SLA, select **Never**, and then go to step [10 \(on page 330\)](#).
5. In the **Calculate time from the start of** section, select a method for calculating your goal and deadline times:

- To start the calculation when a user creates an instance of your case type, select **This case**.
  - To start the calculation when the parent of your case type starts, select **Parent case**.
  - To start the calculation when the top-level parent of your case type starts, select **Top level case**.
6. In the **Goal** section, in the **Days** and **HH:MM:SS** fields, enter the suggested resolution time for a case.
  7. In the **Increase urgency by** field, enter a value by which you want to increase the current case urgency when the goal elapses.
  8. In the **Deadline** section, in the **Days** and **HH:MM:SS** fields, enter the required resolution time for a case.
  9. In the **Increase urgency by** field, enter a value by which you want to increase the current case urgency when the deadline elapses.
  10. Click **Save**.

At run time, your application detects when the goal and deadline are not met, and adjusts the urgency of the assignment. To see how much time you have left to complete a case, refer to the header of the form.

[Building case types \(on page 20\)](#)

[Case types \(on page 28\)](#)

[Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#)

[Defining an escalation action for a case \(on page 335\)](#)

[Completing work on time](#)

## Setting service-level agreements (SLAs) for stages, processes, and steps

Ensure that users complete assignments on time by assigning service-level agreements (SLAs) to stages, processes, and steps. By implementing SLAs, you enforce timely case resolution and ensure that users follow the timeline that you agree on with your customers.

For example, you can define timelines for CSRs to review loan requests within a goal of five days and a deadline of ten business days.

For relevant training materials, see the [Completing work on time](#) module on Pega Academy.

The system calculates goals and deadlines as the time between task assignment and task completion, unless your SLA states otherwise. For more information about advanced SLA configurations, see [Creating a service-level agreement \(SLA\) rule \(on page 332\)](#).

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the upper-right corner of the **Workflow** tab, click **Life cycle**.



3. In the **Case life cycle** section, click the element of a case life cycle for which you want to define an SLA:

- To define a goal and deadline for a stage, click the stage.

**Note:** You cannot set a goal and deadline for the Create stage because the Create stage collects data before case processing starts. For more information, see [The Create stage \(on page 48\)](#).

- To define a goal and deadline for a process, click the process.
- To define a goal and deadline for a step, click the step.

4. In the properties pane, click **Goal & deadline**.

5. In the **Use Service-level agreement (SLA)** list, select the SLA criteria:

- To define custom criteria for the goal and deadline, select **Custom SLA**, and then go to step [6 \(on page 331\)](#).
- To reuse criteria from an existing SLA, select **Existing SLA**. In the auto-complete field, enter the SLA name, and then go to step [8 \(on page 331\)](#).

6. In the **Goal** section, define custom SLA criteria for the goal:

- In the **Goal** section, in the **Days** and **HH:MM:SS** fields, enter the suggested resolution time for a case.
- In the **Increase urgency by** field, enter a value by which you want to increase the current case urgency when the goal elapses.

7. In the **Deadline** section, define custom SLA criteria for the deadline:

- In the **Deadline** section, in the **Days** and **HH:MM:SS** fields, enter the suggested resolution time for a case.
- In the **Increase urgency by** field, enter a value by which you want to increase the current case urgency when the deadline elapses.

8. Click **Save**.

At run time, your application detects when the goal and deadline are not met, and adjusts the urgency of the assignment. To see how much time you have left to complete an assignment, refer to the header of the form.

[Setting service-level agreements \(SLAs\) for case resolution \(on page 329\)](#)

[Completing work on time](#)



## Adjusting initial urgency for an assignment

Communicate the priority of a given task by adjusting initial urgency. By configuring an assignment urgency, you provide a common understanding of task importance. As a result, you optimize workload management so that users can make informed decisions about which task to process next. For example, in a case type for reviewing loan requests from VIP customers, you can increase the initial urgency to ensure that customer service representatives (CSRs) resolve such work as a priority.

Users can check an assignment urgency during run-time task processing. The urgency can increase during a case life cycle when an assignment passes a goal and deadline. The default assignment urgency is 10.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, click a step that has a defined goal and deadline.
3. In the **Step** properties pane, on the **Goal & deadline** tab, open the SLA rule by clicking an **Open** icon next to the rule name.
4. In the **Start of service level** section, in the **Initial Urgency** field, define the initial urgency by entering an integer between 0 and 99.  
An assignment has a default urgency of 10. The system adds the value in this field to determine the initial urgency. For example, if you enter 10, the initial urgency is 20.
5. Click **Save**.

---

### Related information

[Creating a service-level agreement \(SLA\) rule \(on page 332\)](#)

[Assigning tasks to users \(on page 84\)](#)

[Completing work on time](#)

## Creating a service-level agreement (SLA) rule

Enforce service-level agreements (SLAs) and maintain timely resolution of cases by defining the goal, deadline, and passed deadline time intervals. By creating an SLA rule, you can quickly reuse the SLA settings for multiple assignments.

For example, you can notify a manager about a passed goal and deadline, and then transfer the assignment to another user. You then can reuse the setting for multiple cases and assignments, and as a result, manage time and resources efficiently.

For relevant training materials, see the [Completing work on time](#) module on Pega Academy.

Creating an SLA rule in Dev Studio is appropriate for only advanced developers. To take advantage of low-code tools, define SLAs in App Studio. For more information, see [Setting service-level agreements \(SLAs\) for](#)



case resolution (on page 329) and Setting service-level agreements (SLAs) for stages, processes, and steps (on page 330).

1. Create an SLA rule:
  - a. In the navigation pane of Dev Studio, click **Records**.
  - b. Expand the **Process** category, and then click **Service Level Agreement**.
  - c. In the **Label** field, enter a short description for the rule.
  - d. In the **Context** section, select an application to which the rule applies.
  - e. In the **Apply to** field, select a class to which the rule applies.
  - f. In the **Add to ruleset** field, select a ruleset for the rule.
  - g. **Optional:** To enable traceability of the rule, in the **Current work item** section, enter the work item to associate with the rule, for example, a bug ID.
  - h. Click **Create and open**.
2. On the **General** tab, in the **Initial urgency** field, define the urgency for the assignment that you can associate with the rule by entering an integer between 0 and 99.  
An assignment has the default urgency of 10. The system adds the value in this field to arrive at the initial urgency.
3. In the **Assignment ready** list, define the starting time for the assignment:
  - To send the assignment to a worklist or a work queue immediately after the user or an automation creates an assignment, select **Immediately**.
  - To define the starting time by using a Date/Time value, select **Dynamically defined on a property**, and then, in the **Get Date Time From** field, enter the property that references the value.
  - To start the assignment after a period of time, select **Timed delay**, and then, in the **Days**, **Hours**, and **Minutes** fields, specify the delay.
4. In the **Service level definitions** section, in the **Calculate service levels** field, select a method of calculating intervals for goal and deadline:
  - To manually specify the intervals, select **Interval from when assignment is ready**, and then, in the **Goal** and **Deadline** sections, enter the intervals.
  - To define the time intervals as values of properties, select **Set the value of a property**, and then, in the **Goal** and **Deadline** sections, in the **Property** fields, enter the properties that reference goal and deadline dates and times.
5. **Optional:** To facilitate timely resolution of work, define escalation actions that occur when the goal and deadline pass:
  - a. In the **Amount to increase urgency** field, enter the number that an application adds to the initial urgency then the time interval passes.
  - b. In the **Actions** section, in the **Select action** row, click **Edit**.

- c. In the **Perform action** list, select an action that occurs when the time interval passes, and then provide relevant details if necessary.

**Note:** Ensure that the action that you select does not perform the *Commit* method because the commit occurs at the end of SLA rule processing ends. Providing an action that performs the *Commit* might cause issues.

- d. **Optional:** To perform the escalation action only under specific circumstances, in the **When** field, enter a when rule.

An escalation action occurs if the when rule returns a true value.

- e. **Optional:** To add more escalation actions, click **Add an action**, and then repeat steps 5.a (on page 333) through 5.d (on page 334).

**Tip:** You can define escalation actions for goal, deadline, and passed deadline events.

6. **Optional:** To define the number of the passed deadline events and the time when the passed deadline events occur, in the **Passed deadline** section, enter the appropriate values.
7. Click **Save**.

[Completing work on time \(on page 324\)](#)

[Understanding and complying with case management best practices \(on page 26\)](#)

[Managing work across your team \(on page \)](#)

[Pega-ProCom agents \(on page \)](#)

[Service-level agreement rule form \(on page \)](#)

[More about service-level agreements \(on page \)](#)

## Configuring delayed service-level processing

Provide solutions in scenarios when users need to wait before processing an assignment by delaying service-level agreements (SLAs). As a result, you deliver a flexible application and adjust processing to the unique business requirements of your customers.

For example, consider a scenario in which an employee of a trading company has a deadline of two days to review and approve a return of faulty goods from a customer. The deadline count starts when the employee receives the faulty goods, which happens daily at 3:00 PM, even if the review assignment in a

complaint case is ready earlier. In a different scenario, an employee needs to wait for two hours before processing a purchase order, because a customer can cancel finalization of the order for two hours after the purchase.

For relevant training materials, see the [Completing work on time](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, click a step that has a defined goal and deadline.
3. In the **Step** properties pane, on the **Goal & deadline** tab, open the SLA rule by clicking an icon next to the rule name.
4. In the **Start of service level** section, define when an application begins the goal and deadline count by selecting an option from the **Assignment Ready** list:
  - To define the starting time by using a specific *DateTime* property, select **Dynamically defined on a property**, and then in the **Get Date Time From** field, enter the property that references the value.  
  
The *DateTime* property holds a precise starting time, for example, 3:00 PM on Monday August 27.
  - To start the assignment after a specific period of time, select **Timed delay**, and then in the **Days, Hours, and Minutes** fields, specify the delay.
5. Click **Save**.

### Related information

[Understanding and complying with case management best practices \(on page 26\)](#)

[Steps in a case life cycle \(on page 78\)](#)

[Completing work on time](#)

### Defining an escalation action for a case

Ensure that your business process reaches a resolution on time, by defining an escalation action for a case. By enforcing service-level agreements and implementing escalation actions, you enforce timely resolution of cases and meet deadlines that you agree on with your customers.

For example, if a customer service representative (CSR) fails to resolve a case within first goal and then deadline, a reporting manager receives email messages with case details.

For relevant training materials, see the [Completing work on time](#) module on Pega Academy.



1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the case working area, click the **Settings** tab, and then click **Goal & deadline**.
3. In the **Goal** section, in the **Notify** subsection, select recipients of the notification:
  - To notify a creator of the case, select **Creator**.
  - To notify the participants of the case, select **Participants**.
4. If you select to notify participants, click **Manage participants**, select the case participants that you want to notify, and then click **Done**.
5. In the **Message** subsection, select a message type:
  - To send a standard notification, select **Use default message**.
  - To compose an email message, select **Compose a message**, provide a subject, click **Compose**, enter your message, and then click **Done**.
6. In the **Deadline** section, create a notification by repeating steps 3 *(on page 336)* through 5 *(on page 336)*.
7. Click **Save**.

[Defining an escalation action for a stage and a process \*\(on page 336\)\*](#)

[Setting service-level agreements \(SLAs\) for case resolution \*\(on page 329\)\*](#)

### Defining an escalation action for a stage and a process

Ensure that your business use case moves forward in a timely manner by defining an escalation action for a stage and a process in your case life cycle. By notifying interested parties about an elapsed goal or deadline, you can keep the time frames that you agree on for your business use cases.

For example, if an HR worker fails to meet the goal for running a background check on a job candidate, a reporting manager receives a notification. If the worker misses the deadline, the manager receives an email message with relevant details.

For relevant training materials, see the [Completing work on time](#) module on Pega Academy.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, click the element of a case life cycle for which you want to define an escalation action:



- To send a notification after a goal or deadline for a stage that elapses, click the stage.



**Note:** You cannot set an escalation action for the Create stage because the Create stage collects data before case processing starts. For more information, see [The Create stage \(on page 48\)](#).

- To send a notification after a goal or deadline for a process that elapses, click the process.
4. In the properties pane, click the **Goal & deadline** tab.
  5. From the **Use Service-level agreement (SLA)** list, select **Custom SLA**.
  6. In the **Goal** section, in the **Notify** subsection, select recipients of the notification:
    - To notify a creator of the case, select **Creator**.
    - To notify the participants of the case, select **Participants**.
  7. If you choose to notify participants, click **Manage participants**, select the case participants that you want to notify, and then click **Done**.
  8. In the **Message** subsection, select a message type:
    - To send a standard notification, select **Use default message**.
    - To compose an email message, select **Compose a message**, provide a subject, click **Compose**, enter your message, and then click **Done**.
  9. In the **Deadline** section, create a notification by repeating steps [6 \(on page 337\)](#) through [8 \(on page 337\)](#).
  10. Click **Save**.

---

[Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#)

[Defining an escalation action for a case \(on page 335\)](#)

[Completing work on time](#)

### Defining an escalation action for an incomplete assignment

Reduce the risk of unfinished work in a case by defining an escalation action, such as a notification to the manager. The manager receives the notification when an assignment exceeds its goal or deadline.

For example, in a case of reviewing a complaint for purchased goods, an application can notify a manager that a case worker misses a goal for confirming customer details required to pay the reimbursement.

For relevant training materials, see the [Completing work on time](#) module on Pega Academy.



1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the upper-right corner of the **Workflow** tab, click **Life cycle**.
4. In the **Case life cycle** section, click an assignment with a custom goal and deadline.  
For more information, see [Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#).
5. In the step property panel, click **Goal & deadline**.
6. In the **Goal** section, indicate what action to take when an assignment exceeds its goal:
  - To send an email notification, in the **Action** list, click **Notify**, and then select the users to whom you want to send the notification, for example, **Manager**.
  - To reassign the assignment, in the **Action** list, click **Reassign to**, and then specify the user or work queue to whom you want to reassign the assignment.
  - To resolve the case, in the **Action** list, click **Resolve**, and then select the status for the resolved case, for example, **Resolved-Withdrawn**.
7. In the **Deadline** section, define an escalation action by repeating step 6 [\(on page 338\)](#).
8. Click **Save**.

---

[Assignment shapes in processes \(on page 270\)](#)

[Collecting information from a user \(on page 83\)](#)

## Engaging and notifying stakeholders

Communicate with the stakeholders and participants in your Microjourney by sending notifications. Timely notifications enable your stakeholders to act quickly and appropriately to the needs of your business process.

You can better meet your stakeholders' specific needs by ensuring that the notifications that your application sends out include only relevant information. For example, when a stakeholder receives a notification, the text might include a case ID and deadline, so that the stakeholder knows when to expect case resolution. Your application autopopulates these values at run time, saving you time by using predefined templates across multiple cases. To meet your specific business needs, you can also create and send notifications directly from a case type. Configure your application to send notifications only from cases that are relevant to your stakeholders, increasing efficiency by only emphasizing the cases that require their attention.

You can also adjust your solutions to the needs of your stakeholders by defining different channels through which to send your notifications, for example, email, web gadgets, or mobile push notifications.



## Creating a notification

In Dev Studio, you can create a notification to inform users about events that they are involved in or interested in.

1. In the Dev Studio header, click **Create > Process > Notification**.
2. On the Create form, enter values in the fields to define the context of the notification:

### Label

Describes the purpose of the notification.

### Identifier


Specifies a unique identifier for the notification.

You can keep the default value or click **Edit** to define an identifier.

### Add to ruleset

Identifies the name and unlocked version of a ruleset that stores the notification.

3. Click **Create and open** to open the Notifications form.
4. **Optional:** To prevent the notification from being run when referenced, select **Mute notification** on the **Advanced** tab.

 **Tip:** As a best practice, mute overridden notification rules rather than withdrawing the rules.

5. Click **Save**.

---

Notifications (*on page* )

Copying a rule or data instance (*on page* )

Creating a rule specialized by circumstance (*on page* )

## Defining a notification message

In Dev Studio, you can define a notification message that your application sends to users at run time to inform them about an event.

1. Click the Records Explorer and expand the **Process** category.
2. Click **Notification**.
3. Click the notification for which you want to define a message.



4. On the **Notification definition** tab, keep the default field value or enter a field value in the **Message** field that contains your notification message. Click the **Open** icon to create a new field value or to open the field value that you specified.
5. **Optional:** Enter parameters for the field value in the **Parameters** section.
6. Click **Save**.

[Creating a notification \(on page 339\)](#)

[Copying a rule or data instance \(on page \)](#)

[Creating a rule specialized by circumstance \(on page \)](#)

### Adding recipients for a notification

In Dev Studio, you can add recipients for a notification. These recipients are the users who receive the notification at run time.

1. Click the Records Explorer and expand the **Process** category.
2. Click **Notification**.
3. Click the notification for which you want to add recipients.
4. On the **Notification definition** tab, add users who receive your notification at run time in the **Recipients** section:
  - a. In the **Recipient data page** field, enter a data page of list type that contains the recipient properties. Click the **Open** icon to create a new data page or to open the data page that you specified.
  - b. From the **Recipient type** list, select **Operator ID** if the intended recipients are users with valid operator IDs, or users who are part of your application. For external users, select **Email ID**. External users are not part of the application and can receive email notifications only.
  - c. From the **Recipient property** list, enter the property on the data page that holds the value of the recipient **Operator ID** or **Email ID** value.
  - d. **Optional:** To add more recipients from the data page, click **Add property**.
5. **Optional:** To add recipients from another data page, click **Add recipients** and complete the **Recipients** section.
6. Click **Save**.

[Notifications \(on page \)](#)

[Creating a notification \(on page 339\)](#)

[Copying a rule or data instance \(on page \)](#)

[Creating a rule specialized by circumstance \(on page \)](#)



## Notification channels

A notification channel is a carrier that the application uses to notify users about events at run time. For example, users can receive email notifications when posts are made on cases that they are following. You can create notifications for one or all of the available notification channels – web gadget, email, and mobile.

You can define a notification to be sent from a process or an activity and configure the notification delivery method.

You can create notifications for one or all of the available notification channels – web gadget, email, and mobile, or you can create your own custom channel. Web gadget notifications are displayed as messages in the application at run time. You can include additional notification web gadgets to display different types of notifications separately. Users can reply to email notifications to post messages in a conversation. With mobile push notifications, users can approve and reject cases and respond to messages in cases.

Users can control which type of notifications they receive and when they receive them by setting notification preferences.

You can also create custom notification channels if the existing channels do not accommodate your requirements. For example, banks send SMS messages to users to inform them about transactions in their accounts. For information about creating a custom channel, see [Understanding custom Deployment Manager notification channels \(on page 339\)](#) and [Creating custom Deployment Manager notification channels in 5.5.x \(on page 339\)](#). After creating the custom channel, you can configure it on the **Channels** tab for your notification.

You configure notification channels in Dev Studio:

- 
- [Notifications \(on page 339\)](#)
  - [Creating a notification \(on page 339\)](#)
  - [Defining a notification message \(on page 339\)](#)
  - [Adding recipients for a notification \(on page 340\)](#)
  - [Setting notification preferences \(on page 346\)](#)


## Configuring a web gadget notification channel

In Dev Studio, you can configure a web gadget channel to send notifications to users. Web gadget notifications are displayed as messages in the application at run time.

1. Click the Records Explorer and expand the **Process** category.
2. Click **Notification**.
3. Click the notification for which you want to configure a web gadget channel.



4. On the **Channels** tab, in the **Display section** field of the **Web gadget** section, keep the default section or enter a section to display your notification message. Click the **Open** icon to create a new section or to open the section that you specified.

**Note:** To ensure that users receive notifications from at least one channel, you cannot  disable web gadget notifications. However, users can update their notification preferences at run time.

5. From the **Default preference** list, enable or disable the web gadget channel for all the users in your application.  
This value is displayed as the default option for the web gadget channel when users see their notification preferences at run time. They can update the preferences as needed.
6. **Optional:** Configure multiple web gadget channels for different notification categories:
  - a. On the **Advanced** tab, keep the default field value or enter a field value of type `pyNotificationCategory` in the **Category** field that displays your notification under the corresponding web gadget at run time. Click the **Open** icon to create a new field value or to open the field value that you specified.
  - b. Enter parameters for the field value in the **Parameters** section.
7. Click **Save**.

---

[Configuring additional parameters for notification channels \(on page 344\)](#)

[Notification channels \(on page 341\)](#)

[Notifications \(on page \)](#)

[Creating a notification \(on page 339\)](#)

## Configuring an email notification channel

In Dev Studio, you can configure an email channel to send notifications to users. Users can reply to email notifications to post messages in a conversation.

For relevant training materials, see the [Sending emails during case processing](#) module on Pega Academy.

1. Click the Records Explorer and expand the **Process** category.
2. Click **Notification**.
3. Click the notification for which you want to configure an email channel.
4. Perform one of the following actions:

- To customize the email channel, go to [5 \(on page 343\)](#).
  - To disable email notifications for all users, on the **Channels** tab, in the **Email** section, clear the **Enable email** checkbox, and then go to [6 \(on page 343\)](#).
5. If email notifications are enabled, customize the email channel by performing the following actions:
- a. In the **Subject** field, keep the default field value or enter the field value that holds your notification message for email notifications. Click the **Open** icon, to create a new field value or to open the field value that you specified.
  - b. **Optional:** Enter parameters for the field value in the **Parameters** section.
  - c. In the **Correspondence** field, enter a correspondence template for the email message body. Click the **Open** icon, to create a new correspondence or to open the correspondence that you specified.
  - d. From the **Default preference** list, enable or disable the email channel for all the users in your application.  
This value is displayed as the default option for the email channel when users see their notification preferences at run time. They can update the preferences as needed.
6. Click **Save**.

---

[Configuring additional parameters for notification channels \(on page 344\)](#)

[Notification channels \(on page 341\)](#)

[Notifications \(on page \)](#)

[Creating a notification \(on page 339\)](#)

## Configuring a mobile notification channel

In Dev Studio, you can configure a mobile channel to send push notifications to users. Users can approve and reject cases and respond to messages in cases by responding to push notifications.

1. Click the Records Explorer and expand the **Process** category.
2. Click **Notification**.
3. Click the notification for which you want to configure a mobile channel.
4. On the **Channels** tab, in the **Mobile** section, clear the **Enable push notifications** check box to disable mobile push notifications for all users.
5. If mobile push notifications are enabled, customize the mobile channel:

- a. In the **Message** field, keep the default field value or enter the field value that holds your notification message for mobile push notifications. Click the **Open** icon to create a new field value or to open the field value that you specified.
  - b. **Optional:** Enter parameters for the field value in the **Parameters** section.
  - c. From the **Default preference** list, enable or disable the mobile channel for all the users in your application.  
They can update the preferences as needed.
6. Click **Save**.

---

[Configuring additional parameters for notification channels \(on page 344\)](#)

[Notification channels \(on page 341\)](#)

[Notifications \(on page \)](#)

[Creating a notification \(on page 339\)](#)

## Configuring additional parameters for notification channels

In Dev Studio, you can configure additional parameters that the notification channels may require at run time. You can define these parameters as key-value pairs. For example, if you want to notify users that follow a case about a Pulse message in the case, you must add the instance key ( `pzInsKey` ) of the message as an additional parameter. The channels use this parameter to send notifications about the Pulse message at run time.

1. Click the Records Explorer and expand the **Process** category.
2. Click **Notification**.
3. Click the notification for which you want to configure additional parameters for the channels.
4. On the **Advanced** tab, in the **Additional information** section, click the **Add a row** icon to add a parameter.
5. In the **Key** field, enter a key to store the parameter name, for example, `PulseKey`.
6. In the **Value** field, enter a property that holds the value for the specified key, for example, `PostMessage.pzInsKey`. To open the property that you specified or create a new property, click the **Open** icon.
7. Click the **Add a row** icon to add more parameters, and enter values in the **Key** and **Value** fields.
8. Click **Save**.

---

[Notification channels \(on page 341\)](#)

[Notifications \(on page \)](#)

[Creating a notification \(on page 339\)](#)





## Displaying application-specific correspondence

Help users more intuitively send correspondence from cases by providing correspondence prompts that are specific to the context of the current application. As a result, run-time users interact with a relevant list of available correspondence templates, such as templated email messages, so that they can send correspondence faster.

You can mark correspondence as a top-level correspondence rule so that the application prompts this correspondence at run time. By default, applications prompt all top-level correspondence rules that are available for a case through the class hierarchy.

To organize prompts in a more efficient way, create application-specific prompts. For example, in a banking application, you can create a list of correspondence templates that are specific to this application only.

1. On a correspondence rule form, mark correspondence as top-level:
  - a. In the navigation pane of Dev Studio, click **Records**.
  - b. Expand the **Process** category, and then click **Correspondence**.
  - c. In the list of correspondence instances, open the correspondence that you want to edit.
  - d. On the **Prompts** tab, in the **Top level** section, select the **This is a top-level correspondence rule** check box.

Your application displays this correspondence at run time as a prompt.

2. Copy the *ListOfValidCorrs* activity into your application ruleset:
  - a. In the navigation pane of Dev Studio, click **Records**.
  - b. Expand the **Technical** category, and then click **Activity**.
  - c. In the list of activity instances, double-click the **ListOfValidCorrs** activity.
  - d. In the form header, click **Save as**.
  - e. In the **Context** section, in the **Apply to** field, enter your application class.
  - f. In the **Add to ruleset** list, select your application ruleset and ruleset version.
  - g. Click **Create and open**.
3. Edit the *ListOfValidCorrs* activity:
  - a. In the row of the *call LookupList* method, click **Expand to see method parameters**.
  - b. In the **Value** field of the *Action* parameter, enter `classgroup`, as shown in the following figure:

3.  Loop When

**Method Parameters**

Pass current parameter page

Name	Value
* pyObjClass	Rule-Obj-Corr
* pyBaseClass	param.WorkObjectClass
ClassGroup	
* pyListName	List
Action	classgroup
showResults	NO
pageName	pyCorrList
maxRecords	

Editing ListCorrPage activity

c. Click **Save**.

At run time, when users select a correspondence template, a list of prompts includes only top-level correspondence rules from the current application, as shown in the following example:

To

Correspondence prompts

[Notifying participants about events \(on page 235\)](#)

[Sending event notifications from cases \(on page 140\)](#)

[Sending automatic emails from cases \(on page 142\)](#)

## Setting notification preferences

As an application user, you can control the type of notifications that you want to receive and choose the notification channels over which you want to receive them. The available notification channels are gadget, email, and mobile push.

By default, all the notification definitions in an application are classified into the following categories:



- **General** - These notifications are triggered when case-independent events occur, such as a user following another user. Pulse notifications for profile posts, events where a user mentions another user, likes a message that is posted by another user, or follows other users can be set from the rules listed under this category.
- **Case type** - These notifications are triggered when a case-dependent event occurs, such as a case comment being posted. Preferences that you set for case type notifications apply to all the instances of that case type. Notifications for case-related Pulse comments fall under this category and can be configured using the **Comments on cases that I follow** option available under each case. Additionally, you can also specify a different set of preferences for a specific case instance that overrides the case type preferences.

1. Set preferences for general and case type notifications.

a. In the Case Worker portal, click **Profile > Notification Preferences**.

b. For each notification definition under the General or Case type category, set your preference for each channel.

c. **Optional:** Set the frequency of email notifications to one of the following values:

- **Instant** - Email notifications are delivered instantly when they are triggered.
- **Daily** - Email notifications generated over a period of day are consolidated and sent as a daily digest.
- **Weekly** - Email notifications generated over a period of week are consolidated and sent as a weekly digest.
- **Disable** - Email notifications are turned off.

d. Click **Submit**.

2. Set preferences for case type instance notifications.

You can set different notification preferences for a specific case instance by overriding the preferences set for the case type.

a. In the Case Worker portal, open the case type instance.

b. Click **Actions > Notifications**.

c. Click **Override at instance level** to set preferences for this case instance.

d. Update your preferences.



e. Click **Submit**.



**Note:** If you disable the **Receive Notifications** option, all the notifications are stopped including the ones that were set for specific case type instances.

[Configuring Pulse email notifications \(on page 368\)](#)

[Notifying participants about events \(on page 235\)](#)

[Engaging mobile users with push notifications \(on page \)](#)

[Notifications \(on page \)](#)

## Empowering knowledge workers

For dynamic case management, modify your Microjourney to meet the unique needs of your customers by providing users of your application with tools that they can use in specific situations. When you empower knowledge workers with functionalities that they need to dynamically respond to changing situations, you ensure that they can perform relevant actions in every case.

By enriching your case types with optional actions and supporting processes, you ensure that the knowledge workers who use your application, for example customer service representatives (CSRs), have enough data and the right tools to quickly and efficiently process their cases. As a result, you help your customers meet their goals and achieve success, even if the regular path of their case requires additional actions under certain circumstances. Save time and improve the efficiency by displaying supporting and optional tasks only when these actions are relevant.

### Responding to business exceptions in a flow

You can use a ticket in a flow to respond to business exceptions, errors, or events that you define. By moving a case to a specific point in a flow, you can ensure that additional processing occurs so that the case is not left in an incomplete state.

For example, you can activate a ticket to perform cancellation tasks and send correspondence to stakeholders when a user withdraws a case.

1. Click **Configure > Case Management > Work Management > Tickets** to review the tickets that are available in your application and the rules that reference them.
2. Create a ticket.

For more information, see [Creating a ticket \(on page 351\)](#).



3. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
4. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
5. Set the ticket in the flow that defines your exception processing.

- a. On the **Workflow** tab, click **Life cycle**.
- b. Click a process in a stage and then click **Open process**.
- c. On the **Diagram** tab of the Flow form, double-click the shape that is the entry point for your exception processing.

If you do not have exception processing defined, you can add shapes to the flow or create a new flow.

- d. In the **Advanced** section, click **Tickets**.
- e. Click **+ Add ticket**.
- f. In the **Ticket name** field, press the Down Arrow key and select the ticket that your application or a user activates at run time.
- g. **Optional:** To change the label of the ticket in the flow diagram, enter a name in the **Display name** field.
- h. Click **Submit**.
- i. Click **Save**.

6. Activate the ticket in a flow that detects your business exception.

- To automatically activate the ticket:
  - Return to the **Workflow** tab of your case type and click **Life cycle**.
  - Click a process in a stage and then click **Open process**.

You can set a ticket in one flow and raise it in another as long as both flows are in the same case.

- On the **Diagram** tab of the Flow form, add the Utility shape to the flow diagram where the exception occurs.
- Double-click the Utility shape and then configure it to call the `@baseclass.SetTicket` activity with the name of your ticket as an input parameter.

- Click **Submit**.
- Click **Save**.
- To manually activate the ticket:
  - [Flow Actions - Completing the New or Save As form \(on page 303\)](#).  
Ensure that the flow action calls the standard `Work-.pyActionSetTicket` activity during its post processing to activate the ticket.
  - Return to the **Workflow** tab of your case type and click **Life cycle**.
  - Click a process in a stage and then click **Open process**.  
You can set a ticket in one flow and raise it in another as long as both flows are in the same case.
  - On the **Diagram** tab of the Flow form, double-click an Assignment shape to open the **Assignment properties** dialog box.



**Tip:** To make your flow action available for more than one assignment, associate it with a stage in the life cycle of the case type or the case type itself.

- In the **Advanced** section, click **Local actions**.
  - Click **+ Add local action**.
  - In the **Local Action** field, press the Down Arrow key and select the name of your flow action.
  - Click **Submit**.
  - Click **Save**.
7. **Optional:** To change the limits that prevent infinite loops in your flows, override the standard `Pega-ProCom.MaxFlowEnteredCount` and `Pega-ProCom.MaxFlowLoopCount` dynamic system settings in your application.

---

[Changing the path of a process \(on page 280\)](#)

[Adding optional actions to cases \(on page 354\)](#)

[Adding optional actions to stages \(on page 354\)](#)



## Creating a ticket

You can create a ticket to help manage business exceptions or errors in a case.

1. In the header of Dev Studio, click **Create > Process > Ticket**.
2. On the Create form, enter values in the fields to define the context of the ticket.

Options include:

- a. In the **Label** field, enter text that describes the purpose of the ticket.
- b. In the **Apply to** field, press the Down Arrow key and select the class that defines the scope of the ticket.

The class controls which rules the ticket can use. It also controls which rules can call the ticket.

- c. In the **Add to ruleset** field, select the name and unlocked version of a ruleset that stores the ticket.
  - d. To change the default identifier for the ticket, click **Edit**, and then provide a unique value in the **Identifier** field.
3. Click **Create and open** to open the Ticket form.
  4. Click **Save**.

You can make your ticket functional by setting and activating it in different flows in the life cycle of a case.

---

[Responding to business exceptions in a flow \(on page 348\)](#)

[Copying a rule or data instance \(on page \)](#)

[Creating a rule specialized by circumstance \(on page \)](#)

## Policy overrides and suspended work items

Auditors, quality control examiners, and compliance staff can set up rules that detect and automatically halt unusual business exceptions in an application. When desired, they can ensure that work items that record inappropriate situations or processing that violates policies are detected and suspended immediately, regardless of the cause.

The policy override capability supports the detection, tracking, and disposition of such unusual situations. Other work on the work item is blocked until the review completes, and records the an outcome (of Allow or Deny). As with ordinary flows, the complete and detailed work item history of a policy override review flow provides enduring evidence that appropriate procedures were followed.



Business exceptions may occur from many sources — human error, systems failure, software failure, invalid data, unanticipated external situations, or criminal or fraudulent activity. When appropriately designed, your application can detect and suspend processing without any prejudgment of the cause or source. Expert reviewers — operating independently from previously involved parties — can then research, analyze, and remedy problems as appropriate.

They can allow a business exception to be waived, amounting to a case-by-case policy override.

## Examples

A business exception can be defined through a test involving work item property values.

For example, a secured loan amount may be limited to a fixed maximum percent of the value of collateral. In practice, the value of collateral (such as pledged stock holdings) may change from day to day or hour to hour. Changes to collateral or to the loan balance that exceed the limit may occur at any time.

As another example, in most businesses, payments to an employee ordinarily are not to be processed by that employee. While good design may attempt to prevent such situations when a work item and the its parties are initially entered, changes made later — innocently or not — might allow this situation to arise. A test that compares the names of operators who update the case with the names of payees or credit parties is needed.

Finally, business exceptions may be raised based on a pattern of facts that seen individually are less suspicious. Through a computed property in a cover object, conditions in potentially hundreds of member work items can be tallied or summarized. For example, if a dollar limit of \$1000.00 is imposed on each member in a cover, an exception can be raised when more than 50 percent of the members are at that limit amount or close to it.

## Processing scenario

1. To perform an assignment on her worklist, user Mary enters a dollar amount of \$14,000 into an amount field on the flow action form. Mary has complied with all the policies and standards of her job.
2. Separately, in another assignment for another flow executing on the same work item, John updates an account number for a work party. The new account number indicates that the account is closed because of bankruptcy. John also has performed his job responsibilities appropriately.
3. Based on a when condition test in a Declare OnChange rule, a business exception occurs. As a result, the work item status changes to `Pending-PolicyOverride`, and all the currently executing flows are halted.
4. Open assignments for currently executing flows are removed from worklists and work queues; affected users are notified by email.





5. A review flow identified in the Declare OnChange rule starts. The flow execution routes an assignment to a reviewer.
6. While suspended, the work item appears in the Problem Flow report.
7. The reviewer may update the work item, change values, add attachments, and research to reach a verdict.
8. The reviewer records a verdict of `Allow` Or `Deny`.

## Supporting rules

These standard rules support this facility:

- The `Work-.SuspendFlows` activity
- The `Assign-Suspend` class holds assignments for suspended work items. To see a list of assignments, navigate to **Dev Studio > Case Management > Tools > Work Admin > Suspended Work Assignments**.
- The sample flow is `Work-.PolicyOverride`, which calls `Work-.PolicyOverrideFinish`.
- The work item status is `Pending-PolicyOverride`.
- The Policy Overrides item appears in the Overlay menu only for users who hold the `@baseclass.ViewProDex` privilege.

Processing depends on the existence of calendar instances ( `Data-Admin-Calendar` ) for the current year and the time zone of the operator.

The **override** icon does not appear on flow actions for which the Auto-generated HTML box (on the HTML tab) is not selected.

The **override** icon does not appear when the Policy Override is on a harness.

### Related information

How to implement business exception processing with policy overrides (*on page* )

### Configuring and working with optional actions in case types

Provide flexibility for your cases and ensure that case workers, such as customer service representatives, resolve cases that require optional and additional actions that supplement the standard path. For example, you can complement your cases with out-of-sequence actions that do not change the flow of a process but help to provide additional information. You can also define conditions to display a task only when the task is relevant to the case.

[Automating work by creating case types \(on page 27\)](#)

[Adding optional actions to a workflow](#)



## Adding optional actions to cases

Allow customer service representatives (CSRs) to perform out-of-sequence tasks that do not influence the main events in the case life cycle by supplementing your business processes with optional actions.

For example, a customer can give a CSR a new phone number when the CSR processes the Financial History assignment in an Auto Loan case. The CSR can complete the Update Contact Info task to update the phone number without moving the Auto Loan case to a different step or stage.

For relevant training materials, see the [Adding optional actions to a workflow](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, on the **Workflow** tab, click **Optional actions**.
4. In the **Case wide actions** section, click **Action**.
5. In the **Action** menu, select the type of action that you want to add:
  - To add a task, click **Collect information**.
  - To add a process, click **Process**, and then add steps by clicking **Step**.

For more information about configuring steps, see [Steps in a case life cycle \(on page 78\)](#).

- To add a multistep form, click **Multistep form**, and then add steps by clicking **Form step**.
  - To add a user action, click **More > User actions > action name**, and then click **Select**.
6. In the text field, replace the default label for the action with a descriptive action name.
  7. Click **Save**.

At run time, optional tasks appear in the header of a case.

[Case life cycle elements \(on page 43\)](#)

[Adding optional actions to stages \(on page 354\)](#)

[Adding optional actions to paused processes \(on page \)](#)

[Displaying optional actions conditionally \(on page 355\)](#)

## Adding optional actions to stages

Provide customer service representatives (CSRs) with an ability to perform out-of-sequence tasks when your business process enters a specific stage by adding optional actions to stages.

For example, if your business process is to review a job application, you can create a stage Candidate details that includes the Collect personal details, Collect education history, and Collect working experience



processes. If a job candidate gives a CSR a new phone number, the CSR can update the number at any point when processing the stage.

For relevant training materials, see the [Adding optional actions to a workflow](#) module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, on the **Workflow** tab, click **Optional actions**.
4. In the **Stage-only actions** section, under the stage that you want to supplement with an optional action, click **Action**.
5. In the **Action** menu, select the type of action that you want to add:
  - To add a task, click **Collect information**.
  - To add a process, click **Process**, and then add steps by clicking **Step**.

For more information about configuring steps, see [Adding single steps to processes \(on page 82\)](#).

- To add a multistep form, click **Multistep form**, and then add steps by clicking **Form step**.
  - To add a user action, click **More > User actions > action name**, and then click **Select**.
6. In the text field, replace the default label for the action with a descriptive action name.
  7. Click **Save**.

At run time, the action appears in the header of a case when the case enters the stage.

[Case life cycle elements \(on page 43\)](#)

[Adding optional actions to cases \(on page 354\)](#)

[About Flow Actions \(on page 302\)](#)

## Displaying optional actions conditionally

For greater flexibility of your cases, ensure that users, for example customer service representatives (CSRs), see optional tasks only when the tasks are relevant to the case that CSRs currently process. By defining conditions for displaying optional tasks, you accelerate case resolution and save time because users work only on significant tasks. Additionally, you can make optional actions available after the case is resolved. For example, you can help a user correct a typing mistake in an already resolved request.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Case life cycle** section, click **Optional actions**.



- Click an optional action that is a step in the **Case wide actions** section or the **Stage-only actions** section.

If you select an action in the **Case wide actions** section, CSRs can process the actions at any point in the case. If you select the action in the **Stage-only actions** section, CSRs can process the action when a case enters a particular stage.

- In the **User action** section, define the conditions for displaying the task:

Choices	Actions
<p><b>Always display the task</b></p> <p><b>Create a custom condition for displaying the task</b></p>	<p>In the <b>Visible</b> list, select <b>Always</b>.</p> <ol style="list-style-type: none"> <li>In the <b>Visible</b> list, select <b>Custom condition</b>.</li> <li>Next to the list, click the <b>Configure condition</b> icon.</li> <li>In the <b>Configure condition</b> window, select a condition, a comparator, and a value to compare with the condition.</li> <li><b>Optional:</b> To define more conditions, click the <b>Add row</b> icon, and then repeat substep <a href="#">4.c (on page 356)</a>.</li> <li><b>Optional:</b> To group the conditions, select comparators from the list.</li> <li><b>Optional:</b> To reuse the condition for other steps in the future, click <b>Actions &gt; Add to when conditions library</b>.</li> <li>Click <b>Submit</b>.</li> </ol>
<p><b>Use an existing condition for displaying the task</b></p>	<ol style="list-style-type: none"> <li>In the <b>Visible</b> list, select <b>Existing condition</b>.</li> <li>In the list of conditions, select a value.</li> </ol>

- Optional:** To make the action available for resolved cases as well, select the **Allow when case is resolved** checkbox.



**Note:** You can make the action visible exclusively for resolved cases by adding a condition, such as *Is resolved*.

6. Click **Save**.

At run time, when a case reaches an optional action that meets the conditions, the user sees the task. If the action does not meet the conditions, the case skips the task.

[Pausing and resuming processes in cases \(on page 131\)](#)

When condition rules (on page )

## Adding supporting processes to cases

Complement your case life cycle with a supporting process to allow case workers, for example customer service representatives (CSRs), to determine when a case requires additional processing. A supporting process runs independently of the case life cycle, but can change the stage or status of the case. For example, a CSR who works on a hiring process can collect and check additional references from a job candidate, which brings the case closer to a resolution.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, click **Optional actions**.
4. In the **Case wide actions** section, click **Action**.
5. In the action menu, choose a supporting process:
  - To add a new process, click **Process**, and then replace the default label with a new descriptive name.
  - To use an existing process, click **More > Processes > Process name**, and then click **Select**.
6. **Optional:** To start the process when a user creates a case, in the **Process** properties panel, select the **Automatically start when this case starts** check box.
7. Click **Save**.

The supporting process runs as a step in the current stage of the case.

[Adding supporting processes to stages \(on page 358\)](#)

[Displaying supporting processes conditionally \(on page 358\)](#)



## Adding supporting processes to stages

You can provide greater flexibility to your cases and complement them with more information by adding supporting processes to stages. When you add a supporting process to a stage, additional processing can occur only while the case is in that stage. For example, a customer service representative that works on a car insurance case can use a supporting process to collect additional information and materials about the accident, when the case is in a Collect accident details stage.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, click **Optional actions**.
4. In the **Stage-only actions** section, click **Action** below the stage to which you want to add a supporting process.
5. Choose a supporting process:
  - To add a new process, click **Process**.
  - To add an existing process, click **More > Processes**, and then select a process.
6. Replace the default label for the process with a name that describes the purpose.
7. Click **Save**.

At run time, you can refer to the header of a case to find the supporting processes that are attached to the current stage.

[Processes in a case life cycle \(on page 66\)](#)

[Adding optional actions to stages \(on page 354\)](#)

[Steps in a case life cycle \(on page 78\)](#)

## Displaying supporting processes conditionally

To save time during case processing, ensure that users, such as customer service representatives (CSRs), see only the actions that are relevant to a case by defining conditions for displaying supporting processes. For example, a CSR might need to resolve a supporting process only when the case is blocked, to provide more details to bring the case closer to resolution.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, click **Optional actions**.



4. Click a supporting process, which is a sequence of steps in either the **Case wide actions** section or the **Stage-only actions** section.

If you select a process in the **Case wide actions** section, CSRs can see the process at any point in the case. If you select the process in the **Stage-only actions** section, CSRs can process the action when a case enters a particular stage.

5. In the **Process** properties panel, define the conditions for displaying the supporting process:

Choices	Action
<b>Always display the process</b>	In the <b>Visible</b> list, select <b>Always</b> .
<b>Use a when rule for displaying the process</b>	<ol style="list-style-type: none"> <li>In the <b>Visible</b> list, select <b>When rule</b>.</li> <li>In the field that appears, press the Down arrow key, and then select a when rule that you want to use.</li> </ol>
<b>Use an expression for displaying the process</b>	<ol style="list-style-type: none"> <li>In the <b>Visible</b> list, select <b>Expression</b>.</li> <li>In the field that appears, press the Down arrow key, and then select an expression that you want to use.</li> </ol>
<b>Create an expression for displaying the process</b>	<ol style="list-style-type: none"> <li>In the <b>Visible</b> list, select <b>Expression</b>.</li> <li>Next to the list, click the <b>Build an expression</b> icon.</li> <li>In the <b>Expression builder</b> dialog box, define a Boolean expression that evaluates property values, functions, or when conditions. For more information about building expressions, see Building expressions with the Expression Builder (on page ).</li> </ol>

6. In the **Process** properties panel, define the conditions for displaying the supporting process:

Choices	Actions
<b>Always display the process</b>	In the <b>Visible</b> list, select <b>Always</b> .
<b>Create a custom condition for displaying the process</b>	<ol style="list-style-type: none"> <li>In the <b>Visible</b> list, select <b>Custom condition</b>.</li> <li>Next to the list, click the <b>Configure condition</b> icon.</li> </ol>

Choices	Actions
	<ul style="list-style-type: none"> <li>c. In the <b>Configure condition</b> window, select a condition, a comparator, and a value to compare with the condition.</li> <li>d. <b>Optional:</b> To define more conditions, click the <b>Add row</b> icon, and then repeat substep <a href="#">6.c (on page 360)</a>.</li> <li>e. <b>Optional:</b> To group the conditions, select comparators from the list.</li> <li>f. <b>Optional:</b> To reuse the condition in the future, click <b>Actions &gt; Add to when conditions library</b>.</li> <li>g. Click <b>Submit</b>.</li> </ul>
<b>Use an existing condition for displaying the process</b>	<ul style="list-style-type: none"> <li>a. In the <b>Visible</b> list, select <b>Existing condition</b>.</li> <li>b. In the list of conditions, select a value.</li> </ul>

7. Click **Save**.

At run time, when a case reaches a supporting process that meets the conditions, the user sees the process.

[Adding supporting processes to cases \(on page 357\)](#)

[Adding supporting processes to stages \(on page 358\)](#)

[When condition rules \(on page \)](#)

### Preparing for collaboration with users by using Pulse

Improve collaboration and conversation among users within a specific context, such as a case, by adding the Pulse collaboration gadget to your application. By sharing information in Pulse, users can work together to resolve their work more quickly. In Pulse, users can post, view, and reply to messages, to promptly exchange information. To enrich messages, users can attach content to posts and style the text. For example, users can bold important information within the message. Ensure the best-tailored experience when using Pulse by defining default feed sources, so that users immediately see only the relevant information. By configuring a message display, you define what elements a message shows, so



that users find all the vital information in the Pulse posts. For instance, you can define a message title to include a link to a case.

You can use Pulse to collaborate in end-user portals when you work on a case, update your user profile, or view your activity feed. Because Pulse is a reusable tool, you can embed it in other places in your application. For improved communication in a case context, you can configure Pulse for different case types, so that users collaborate to reach a case resolution faster. You configure Pulse for a case type directly in Case Designer.

**Note:** To configure Pulse for case types, ensure that your application includes UI Kit 15 or newer.

To post and view messages in Pulse, you embed the *pxFeed* gadget in your application. The *pxFeed* gadget shows message replies, messages in cases that you follow, messages that reference users, profile messages, bookmarked messages, and application messages. Include this gadget within a section or harness in your application to view the activity feed, for example, in the dashboard of the Case Manager portal. You can use the gadget to display the feed within a specific context, such as a case, a team, or a custom context. You can also configure the Pulse gadget so that users can post messages directly to the activity feed, or to show posts from internal or external data sources, such as the following data page APIs:

*D\_pxPostRepliesFeed*

Retrieves replies to user posts

*D\_pxFollowedCasesFeed*

Retrieves posts to cases followed by users

*D\_pxMentionsFeed*

Retrieves posts that reference users

*D\_pxProfileFeed*

Retrieves posts to user profile pages

*D\_pxBookmarkFeed*

Retrieves posts that users bookmark

*D\_pxPosts*

Retrieves posts for a case, rule, custom, or operator context.

## Examples of collaboration in Pulse

The following scenarios are examples of how you can use Pulse to increase collaboration:



- Customer service representatives and product managers that resolve customer issues
- Account executives that track a sales opportunity
- Marketing personnel that discuss a campaign design
- Financial services professionals that approve a loan or credit card request

Collaborate on cases by completing the following tasks:

---

### Related information

[Collaborating with users by using Pulse \(on page 499\)](#)

[Sourcing attachments from external storage \(on page 392\)](#)

## Adding the Pulse gadget to your application

Provide a collaboration tool for users of your application by adding the Pulse gadget to your application. By posting, viewing, and commenting on messages, users can work together to resolve cases faster. Users that post Pulse messages can make their posts more meaningful by adding attachments, including links, or by formatting the text.

1. In the navigation pane of Dev Studio, click **Records**.
  2. Expand the **User Interface** category, and then click **Section**.
  3. Open a section in which you want to add a Pulse gadget.
  4. Add the Pulse gadget:
    - If you edit the section in the View editor, click **Add > Data display > Embedded section**.
    - If you use a full section editor, click **Structural**, and then drag **Embedded section** to a place in the layout where you want to display Pulse.
  5. In the **Section include modal** window, in the **Section** field, enter `pxFeed`.
  6. Click **Submit**.
  7. Click **Save**.
- 

[Collaborating with users by using Pulse \(on page 499\)](#)

[Collaboration with users by using Spaces \(on page 510\)](#)

## Enabling users to post messages in the activity feed

Enable users to post messages in an activity feed for a specific context. For example, a user can make release announcements in the context of an application for all the application users to see. Users can view their activity feeds in the Case Manager and Case Worker portals when they click the **Pulse** menu option.

This procedure applies to standard Pega Platform applications. For applications based on Cosmos React, see [Configuring Pulse feed sources for Cosmos React \(on page 373\)](#).



1. In Dev Studio, search for the `pyPulseWrapper` section.
2. Override the `pyPulseWrapper` section that includes the `pxFeed` section.
3. On the **Design** tab of a section or harness, click the `pxFeed` section.
4. Open the Layout Properties panel by clicking the **Settings** icon.
5. On the **Parameters** tab, in the **Configure post settings** section, select the **Enable post** checkbox.
6. In the **Context key** field, define the context for posting messages by entering a key value.
7. In the section, select the message types that users can post:
  - To allow users to post public messages, select **Message**.
  - To allow users to post private messages, select **Private post**.
  - To allow users to create and assign tasks, select **Task**.

The post types are displayed in a menu when you open Pulse and click **Post**. For more information, see [Posting a message in Pulse \(on page 499\)](#).

8. **Optional:** Update the description of a selected message type:
  - a. Next to the message type, click the **Edit** icon.
  - b. In the **Feed name** field, update the description.
  - c. Click **Done**.
9. Click **Submit**.
10. Click **Save**.

[Collaborating with users by using Pulse \(on page 499\)](#)

[Creating an embedded section \(on page \)](#)

[Creating feed sources for activity feeds \(on page 363\)](#)

[Adding the Pulse gadget to your application \(on page 362\)](#)

## Creating feed sources for activity feeds

Inform users about events that are relevant to them by creating feed sources for messages from internal or external data sources. After you create a feed source, you can include it in a user activity feed. For example, you can configure a Pulse feed to display posts from members of different departments.

This procedure applies to standard Pega Platform applications. For applications based on Cosmos React, see [Configuring Pulse feed sources for Cosmos React \(on page 373\)](#).

1. In the header of Dev Studio, click **Create > Process > Pulse Feed**.
2. In the **Pulse feed record configuration** section, in the **Label** field, enter a short description for the feed source that you want to create.
3. In the **Apply to** field, select a class to which this feed source applies.
4. In the **Add to ruleset** field, select the name of the ruleset that contains the feed source.
5. Click **Create and open**.



6. In the **Feed label** field, enter a description for the new feed.  
By default, the system populates the feed label with the label that you provide in the **Label** field in step 2 (on page 363).
7. In the **Data page** field, enter the data page that retrieves the messages for the feed.
8. In the fields that appear based on the data page that you choose, configure additional parameters.
9. Click **Save**.

[Collaborating with users by using Pulse \(on page 499\)](#)

[Creating an embedded section \(on page \)](#)

[Enabling users to post messages in the activity feed \(on page 362\)](#)

[Adding the Pulse gadget to your application \(on page 362\)](#)

## Adding feed sources to activity feeds

Ensure that users can view relevant messages by adding a feed source to their activity feed. For example, you can add an external feed source, so that customer service representatives (CSRs) that work with finances can see posts about currency rates.

This procedure applies to standard Pega Platform applications. For applications based on Cosmos React, see [Configuring Pulse feed sources for Cosmos React \(on page 373\)](#).

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **User Interface** category, and then click **Section**.
3. Open a section that contains the Pulse gadget.
4. In the *pxFeed* section, click **Edit this section**.
5. In the **Cell Properties** window, click the **Parameters** tab.
6. In the **Configure additional feed sources** section, click **Add feed source**, and then specify the feed source:
  - a. In the **Name** field, enter the name of the feed source that you want to add.
  - b. **Optional:** To display messages from the feed source every time a user opens the Pulse gadget, select the **Display feed by default** check box.
  - c. Click **OK**.
7. Click **Submit**.
8. Click **OK**.

[Collaborating with users by using Pulse \(on page 499\)](#)

[Configuring Pulse for case types \(on page 220\)](#)

[Configuring display of Pulse messages \(on page 366\)](#)

[Posting a message in Pulse \(on page 499\)](#)



## Determining default feed sources

Ensure that users see only relevant posts by specifying which feed sources Pulse displays by default.

This procedure applies to standard Pega Platform applications. For applications based on Cosmos React, see [Configuring Pulse feed sources for Cosmos React \(on page 373\)](#).

Users can override the default settings at run time by filtering messages; however, the default changes are restored after reloading the page.

1. In the header of Dev Studio, search for the section that contains the *pxFeed* section that you want to configure, for example, *pyPulseWrapper*.
2. On the **Design** tab of the section or harness, click the *pxFeed* section.
3. Open the **Layout Properties** panel by clicking the **View properties** icon.
4. In the **Layout Properties**, click the **Parameters** tab.
5. Next to the feed source that you want to mark as the default source, select the **Default** checkbox.
6. Click **Submit**.

Users can see messages from default feed sources.

[Collaborating with users by using Pulse \(on page 499\)](#)

[Enabling users to post messages in the activity feed \(on page 362\)](#)

## Configuring Pulse for case types

Provide case workers, such as customer service representatives (CSRs), with a collaboration tool for open discussion of their work, by configuring Pulse. To accelerate case type development, configure Pulse directly in the case type settings. When you enable Pulse, case workers can post messages to their colleagues and receive replies. To ensure that case workers only see relevant messages, you can define default feed sources and post display conditions.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the case working area, click the **Settings** tab.
4. In the **Pulse** section, configure the Pulse settings:

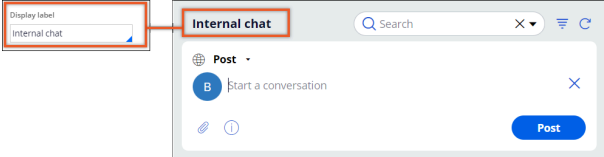
### Choices

**Change the default label**

### Actions

In the **Display label** field, enter a new label.



Choices	Actions
	<p>For example, enter <code>Internal chat</code>, as in the following figure:</p>  <p>Setting the Pulse gadget label for a case</p>
<p><b>Enable posting Pulse messages</b></p>	<p>Select the <b>Allow users to add posts</b> check box.</p>
<p><b>Define a condition for post visibility</b></p>	<p>In the row of a post type that you want to configure, in the <b>Visibility</b> list, select when the application displays posts.</p>
<p><b>Add a feed source</b></p>	<ol style="list-style-type: none"> <li>In the <b>Configure additional feed sources</b> section, click <b>Add Pulse feed</b>.</li> <li>In the <b>Pulse feed name</b> list, select the source that you want to add.</li> <li><b>Optional:</b> To display the feed source default, select the <b>Display feed by default</b>.</li> <li>Click <b>OK</b>.</li> </ol>

5. Click **Save**.

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

[Collaborating with users by using Pulse \(on page 499\)](#)

## Configuring display of Pulse messages

To ensure that users see relevant information about a Pulse message, such as an icon or the profile image of the users posting and replying to messages, customize the message display. For example, you can configure the messages that your application posts automatically to display your company logo and a case ID as a title of the message.

- In the navigation pane of Dev Studio, click **Records**.
- Expand the **Process** category, and then click **Pulse Feed**.
- Open a Pulse feed that you want to edit.
- In the **Feed display settings**, in the **Icon** list, select the type of icon that you want to display in the message:

Choices	Actions
<b>Display a user profile image</b>	<ol style="list-style-type: none"> <li>In the <b>Icon</b> list, select <b>User</b>.</li> <li>In the <b>User reference</b> field, enter a property that references the user whose profile image you want to display</li> </ol>
<b>Display an icon that a system provides</b>	<ol style="list-style-type: none"> <li>In the <b>Icon</b> list, select <b>Icon class</b>.</li> <li>Next to the <b>Icon class</b> field, click the <b>Open the icon class picker</b> icon.</li> <li>In the <b>Icon class picker</b> window, select an icon that you want to display next to the Pulse message.</li> </ol>
<b>Display an image from the Image Catalog</b>	<ol style="list-style-type: none"> <li>In the <b>Icon</b> list, select <b>Image</b>.</li> <li>Next to the <b>Image reference</b> field, click the <b>Show Image Viewer</b> icon.</li> <li>in the <b>Search</b> field, enter a key word or the name of the image that you want to use, and then click <b>Find</b>.</li> <li>Select the image.</li> </ol> <div data-bbox="862 1205 1463 1482" style="background-color: #e1f5fe; padding: 10px; border: 1px solid #bbdefb;"> <p><b>Note:</b> Only standard Pega Platform applications support adding images from the image catalog. Applications based on Cosmos React do not provide this option.</p> </div>

5. In the **Title** list, select a display format for the title of the message:

- To display the title as a value, in the **Title** list, select **Property**, and then in the **Property reference** field, enter a property that stores the value that you want to use.

For example, to display a case label as a title, enter `.pyLabel1`.

- In a standard Pega Platform application, to display the title as a section, in the **Title** list, select **Section**, and then in the **Section reference** field, enter a property that references the section that you want to use.
  - In an application based on Cosmos React, to use more complex text for the title, for example a sentence, in the **Title** list, select **Field value**, and then in the **Field value reference** field, enter a property that references the field value that you want to use.
6. In the **Time** field, enter the property that retrieves the time at which a user adds the message.
  7. For a standard Pega Platform application, in the **Message type** field, select a display format for the message:
    - To display a message as plain text, select **Message**, and then in the **Property reference**, enter a custom property.
    - To display a message as a section, select **Section**, and then in the **Section reference**, enter a property that references a custom section.
  8. For an application based on Cosmos React, in the **Message reference** field, enter the name of a view that you want to use for displaying the messages.  
For more information about views in Cosmos React, see *Working with views (on page 507)*.
  9. **Optional:** To allow users to comment on a message, in the **Comments** section, select the **Allow comments** check box, and then, enter the custom property from the data page that uniquely identifies the feed item.  
The **Allow comments** check box is available only for some Pega-delivered feed source types, such as *pxMySpacesFeed*, which displays Pulse messages from spaces in which a user is a member.
  10. Click **Save**.

---

[Viewing your activity feed \(on page 507\)](#)

[Posting a message in Pulse \(on page 499\)](#)

[Collaborating with users by using Pulse \(on page 499\)](#)

## Configuring Pulse email notifications

Inform users in real-time about updates to their Pulse conversations by sending email notifications. In Pulse, you can choose to receive emails when another user references you, likes your messages, posts on your profile, posts comments on a conversation in which you participate, or posts a comment on a case that you follow. As a result, you enhance communication and speed up case resolution. For example, when team members have a Pulse conversation about preparing a loan offer for a VIP customer, each member can receive email notifications about any activity in the conversation. To save time, users can also reply to Pulse messages by sending an email.

Receiving Pulse notifications by email helps users resolve cases more quickly, because users can exchange information directly from their email inboxes without logging in to an application. In the example of preparing a loan offer, a user can receive email notifications when:





- Other users post Pulse messages in the case.

If the message includes an attachment, the user receives the attachment in the email.

- Other users reference the email recipient in a Pulse message.
- Other users like a message sent by the email recipient.

To resolve the case more quickly, the user can respond to Pulse messages by replying directly to the Pulse notification from their inbox, without logging in to an application. Pulse then displays the response as a message in the conversation. To provide more context and information, the user can include an attachment in their email, that an application attaches to the Pulse message.

You can enable users to receive and respond to Pulse email notifications by performing the following actions:

- 
- Integrating your application with an email provider (*on page* )
  - [Creating a team \(on page 492\)](#)
  - Operators (*on page* )
  - [Collaborating with users by using Pulse \(on page 499\)](#)

## Configuring an email account for Pulse notifications

To notify users about any activity in their Pulse conversations by email, configure an email account to process Pulse notifications. As a result, users can immediately know about new Pulse messages in cases that they follow, replies to their messages, and when other users reference them in Pulse comments. Immediate exchange of information can help users resolve work faster. For example, if a customer service representative (CSR) posts a Pulse message with a question about the maximum loan amount that a customer can receive, and references a manager in the message, the manager receives an email that contains the message and can immediately respond to the question.

1. In the header of Dev Studio, click **Create > Integration-Resources > Email Account**.
2. In the **Short description** field, briefly describe the purpose of the email account.
3. In the **Account Name** field, enter `PulseNotifications`.

**Note:** You can have only one account with the name `PulseNotifications` in the system.

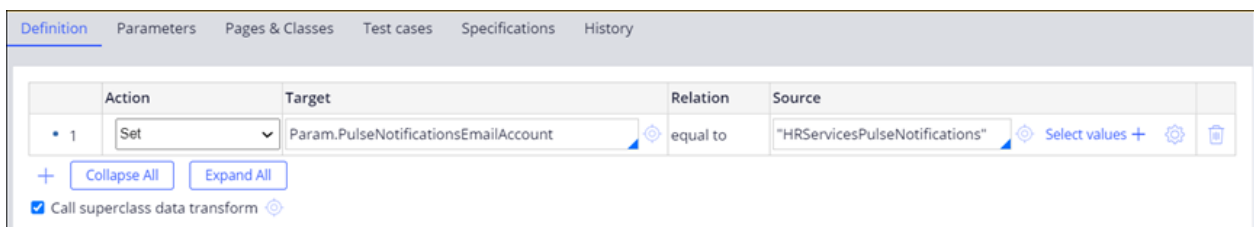
4. Click **Create and open**.
5. Configure the email account as appropriate.  
For more information, see [Configuring outbound email in App Studio \(on page \)](#).
6. Click **Save**.



## Configuring Pulse notifications on additional email accounts

Enhance and categorize communication in your organization by creating multiple email accounts to send email notifications about Pulse messages. Consequently, users can quickly understand the context of the message, and as a result, process their work faster. For example, you can configure an email account, such as HRServices@example.com, to send Pulse emails from the HR Services application, or SalesAutomation@example.com, to send Pulse emails from the Sales Automation application, so that users can quickly identify which application sends the notification.

1. Copy the `pyPulseNotificationsEmailAccount` data transform that belongs to the *Pega-Social* ruleset, to your application ruleset:
  - a. In the navigation pane of Dev Studio, click **Records**.
  - b. Expand the **Data Model** category, and then click **Data Transform**.
  - c. Click **Create**.
  - d. In the **Label** field, enter a new name for the rule.
  - e. In the **Add to ruleset** list, select your application ruleset.
  - f. Click **Create and open**.
2. On the **Definition** tab, in the **Source** column for the `Param.PulseNotificationsEmailAccount` target, enter your email account name, as in the following figure:



Data transform for additional email accounts

3. Click **Save**.

## Enabling users to respond to Pulse email notifications

Enhance communication and accelerate work processing by enabling an option to reply to Pulse messages by email. When users can post Pulse messages without logging in to an application, information exchange is faster and more convenient, and as a result, cases reach resolution faster. For example, in a sales automation scenario, a Pulse discussion occurs about converting a prospect to a customer in a case. A sales person can quickly add comments to a post and attach a related document by replying to an email notification, without logging in to the application.

Users can receive Pulse notification emails in cases that they follow and for which they choose to receive notifications. Users also receive emails when other users reference them in Pulse posts, or like their posts. You can allow users to quickly post attachments or text as replies by responding to emails instead of logging in to the application.



When users reply to emails, Pulse immediately fetches replies from Outlook, Gmail, Yahoo, iPhone, and Android email clients.



**Note:** Users cannot post replies to Pulse by responding to the email notifications that they receive when other users like their messages.

To enable users to respond to Pulse email notifications, you need to create a relevant service package in your application, copy an existing service email rule and adjust it to your needs, and then create an email listener.

1. Create a service package in the application ruleset:
  - a. In the header of Dev Studio, click **Create > Integration-Resources > Service Package**.
  - b. In the **Short description** field, briefly describe the purpose of the service package.
  - c. In the **Service Package Name** field, enter a label for the service package.
  - d. Click **Create and open**.
  - e. In the **Context** section, in the **Processing mode** list, select an option that matches your application, depending on whether your application is stateless or stateful.
  - f. In the **Service access group** field, enter the access group of your application.  
Ensure that the access group that you want to use can access service methods.
  - g. Clear the **Requires authentication** checkbox to bypass authentication and allow Pega Platform to invoke service methods.
  - h. To use TLS/SSL for service REST rules that belong to this service package, select **Require TLS/SSL for REST services in this package**.  
For more information, see *Defining processing and authentication for service packages (on page )*.
  - i. Select the **Suppress Show-HTML** checkbox.
  - j. Click **Save**.
2. Create a service email rule by copying an existing rule:
  - a. In the header of Dev Studio, click **Configure > Integration > Services > Service Definitions**.
  - b. Expand **Service Email in package PulseEmailService class PegaSocial-Message**.

- c. Click **pzCreatePulseReply**.
- d. Click **Save as**.
- e. In the **Label** field, enter a new name for the rule.
- f. In the **Customer Package Name** field, enter the name of the service package that you created in step 1 (*on page 371*).
- g. In the **Customer Class Name** field, enter a service class name.
- h. In the **Add to ruleset** list, select your application ruleset.
- i. Click **Create and open**.
- j. In the **Primary Page** section of the **Service** tab, ensure that the **Primary page class** field stores the `PegaSocial-Message` value.
- k. In the **Service activity** section, ensure that the **Activity name** field stores the `pzPostReplyFromMail` value.
- l. On the **Request** tab, ensure that the **Message header** and **Message data** sections store the same values as the `pzCreatePulseReply` service email rule.
- m. On the **Response** tab, ensure that the value of the **Message type** list is `None` to avoid receiving blank emails when users reply to Pulse emails.
- n. Click **Save**.

### 3. Create an email listener rule:

- a. In the header of Dev Studio, click **Create > Integration-Resources > Email Listener**.
- b. In the **Short description** field, describe briefly the purpose of this listener.
- c. In the **Listener Name** field, enter a label for the listener.
- d. Click **Create and open**.
- e. On the **Properties** tab, from the **Startup option** list, select the appropriate option for running the listener, depending on your requirements.  
For more information about creating email listeners, see *Creating an email listener (on page )*.
- f. In the **Email Account** field, enter the email account for the listener that you use for sending emails with Pulse notifications.

- g. In the **Service information** section, in the **Service package** field, enter the service package that you created in step 1 ([on page 371](#)).
- h. In the **Service class** field, enter the name of the service class name that you entered in the **Customer Class Name** field in 2 ([on page 371](#)).
- i. In the **Service method** field, enter the name of the service email that you created in 2 ([on page 371](#)).
- j. On the **Processes** tab, ensure that the **No Attachments** checkbox is cleared, so that attachments can be sent to email replies.
- k. Click **Save**.

## Configuring Pulse feed sources for Cosmos React

Ensure that users of your Cosmos React applications can view relevant messages by adding a feed source to their activity feed. For example, you can add an external feed source, so that customer service representatives (CSRs) that work with finances can see posts about currency rates.

1. In the navigation pane of Dev Studio, click **App**.
2. Click **Classes**.
3. In the text field, enter `Rule-Pulse`, and then open the *Rule-Pulse* rule.
4. On the list of Pulse instances, open the instance for which you want to edit the feed sources:
  - To edit how Pulse displays in your application dashboard, open *pyDashboardFeed*.
  - To edit how Pulse displays in cases, open *pyCaseFeed*.
5. **Optional:** To give the Pulse gadget a custom name in your application, in the **Display label** field, enter a label.
6. Select the **Allow users to add posts** check box.
 

If you leave the check box clear, users can view Pulse posts, but cannot post any messages.
7. If users can add Pulse posts, configure post visibility:
  - a. In the **Context Key** field, enter the context in which users can add their posts.
  - b. In the list of post types, for each post type, in the **Visibility** section, select when you want posts of this type to display in the Pulse gadget:
    - To make the messages always visible, select **Always**.
    - To make the messages visible only in specified circumstances, select **Existing condition**, and then enter a condition.
    - To hide the messages, select *Never*.

8. **Optional:** To add more feed sources, in the **Configure additional pulse feeds** section, click **Add pulse feed**, and then specify the feed source:
  - a. In the **Name** field, enter the name of the feed source that you want to add.
  - b. **Optional:** To display messages from the feed source every time a user opens a view that includes the Pulse gadget, select the **Display pulse feed by default** check box.
  - c. Click **OK**.
9. Click **Save**.

---

[Configuring Pulse for case types \(on page 220\)](#)

[Configuring display of Pulse messages \(on page 366\)](#)

[Posting a message in Pulse \(on page 499\)](#)

[Collaborating with users by using Pulse \(on page 499\)](#)

## Integrating cases with external resources

Expand your Microjourney to take advantage of functionalities outside of your application, by connecting cases to external resources. By expanding your processing to external resources, you can significantly speed up case resolution.

For relevant training materials, see the [Integrating with external applications](#) module on Pega Academy.

For example, you can increase the functionality of your application by assigning tasks to an external application, or to a robotic queue.

### Integrating a case with an external resource

You can call an external resource, such as a robotic automation or another application, from a flow. By reusing the services that other resources provide, you can enhance your cases without having to maintain extra functionality in your application.

The following tasks can help you call an external resource from a flow:

---

[Creating a stand-alone process \(on page 249\)](#)

[Calling one process from another process \(on page 292\)](#)

### Assigning a task to a robotic work queue

Integrate Pega Robotic Process Automation™ (RPA) with your cases to define tasks that a virtual machine (VM) can do. As a result, you can reduce the need for human input in your application and resolve cases faster.



Consider the example of a case type for requesting a new credit card. A customer service representative (CSR) needs to pause the call, log in to a system for ordering the printing of a new credit card, then switch to a system for ordering the shipment of the newly created card. These operations might take considerable time and attention from the CSR, and are prone to human error. However, the CSR can assign these tasks to a robotic work queue, providing quicker, more accurate service to the customer.

You assign a task to a robotic work queue by using the **Assign to robot queue** shape in a process.

For relevant training materials about using RPA, see the [Pega Robotic Process Automation](#) module on Pega Academy.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the process in which you want to add a robotic assignment, click **Add step to the process > More > Automations > Assign to robot queue**, and then click **Select**.
3. In the **Step** properties pane, on the **General** tab, in the **Route to queue** field, enter the name of the work queue from which the VM retrieves assignments.

The **Assign to robot queue** shape creates the assignment that the VM retrieves.

4. In the **Run robotic automation** field, enter the automation that the virtual machine runs after retrieving the assignment.  
For more information, see [How do I configure Robotic Automation for Pega Platform applications?](#).
5. **Optional:** To update the status of a case by synchronizing it with the state of the robotic assignment, click **Set case status**, and then, in the **Set case status** dialog box, for each of the assignment states that you want to map to a case status, perform one of the following actions:
  - To select a status that a case enters when a robotic assignment reaches a certain state, press the Down arrow key in the corresponding field.
  - To create a custom status, enter a unique name in the field, and then click the **Open rule** icon.



**Note:** If you do not select a status in the corresponding field, the case keeps its previous state.

The following predefined robotic assignment states add the capability to automatically update your case status and track its progress in a more granular way:

Robotic assignment state	Description
<b>Assignment is queued</b>	Change case status when a robotic assignment is added to a work queue.

Robotic assignment state	Description
	<p><b>Note:</b> Work queues are called <i>assignment types</i> in Pega Robot Manager.</p> <p><b>i</b> For more information, see Pega Robot Manager as a control center for your robotic workforce (<i>on page</i> ).</p>
<b>Robotic Automation is in progress</b>	Change case status when a robot pulls the assignment and starts working on it.
<b>Robotic Automation has conflicting results</b>	Change case status when a robotic assignment is completed but returns data that fails the validations that are specified for the case.
<b>Robotic Automation fails</b>	Change case status when a robotic assignment cannot be completed and fails.
<b>Robotic Automation has timed out</b>	Change case status when a robotic assignment cannot be completed in the threshold time that is specified in the assignment type (work queue).

6. **Optional:** To validate the incoming values after your automation is complete, in the **Assignment validation criteria** field, select a validation mode:

Choices	Actions
<p><b>Use an existing validation rule</b></p>	<p>a. Select <b>Existing validation</b>.</p> <p>b. In the list that appears below the <b>Assignment validation criteria</b> field, select a validation rule.</p> <p><b>Note:</b> You can select only a validation rule that is a relevant record.</p> <p><b>i</b> For more information, see Relevant records for rule reuse (<i>on page</i> ).</p>



Choices	Actions
<b>Use custom validation</b>	<ol style="list-style-type: none"> <li>Select <b>Custom validation</b>.</li> <li>Click the <b>Click to configure conditions</b> icon.</li> <li>In the <b>Configure condition</b> window, specify how your application validates the automation output.</li> </ol>

If the validation fails, the system routes the assignment to the **Conflicting Robotic Assignments** work queue. If an assignment fails, the system routes the assignment to the Failed Robotic Assignments work queue.

- Optional:** To apply a goal and deadline to the robotic assignment step, on the **Goal & deadline** tab, in the Assignment service level field, press the Down arrow key, and then select the name of a service-level agreement.

For more information, see [Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#).

- Close the **Robot queue** dialog box by clicking **Submit**.
- Click **Save**.

---


[Case life cycle elements \(on page 43\)](#)

[Robotic process automation \(on page \)](#)

[Configuring a work queue for robotic automation \(on page \)](#)

## Assigning a task to an external application

You can use the Assignment Service shape to send a request from a process to an external application, and wait for the application to respond. For example, in a mortgage case, a loan officer can assign a task to an external party to get the credit score for a customer, and proceed with reviewing a loan request from the customer only when the external party sends the credit score.

The Assignment Service shape  identifies an activity that connects your instance of Pega Platform™ to an external system to send and receive data.

The Assignment Service shape moves a case to the next step only after receiving a response from the external system. The Assignment Service tasks support asynchronous coordination with external systems, in contrast to Integrator tasks, which support synchronous connections.

For more information, see [Connecting a process to an external application \(on page 380\)](#) and [Asynchronous integration](#).



1. Add the Assignment Service shape to your process.
  - a. In the navigation pane of Dev Studio, click **App**.
  - b. On the **Classes** tab, expand the case type for which you want to add the assignment, and then click **Process > Flow**.
  - c. In the list of process instances, double-click a process that you want to open.
  - d. On the **Flow** form, on the **Diagram** tab, click the **Flow shapes** icon, and then click **Advanced Shapes > Assignment Service**.
  - e. Drag the Assignment Service shape to a position on the flow diagram based on the order of events in the process.
  - f. Connect the Assignment Service shape by dragging connector end points to connection points on different shapes in the process.
2. Modify the Assignment Service shape to send a request from a process to an external application.
  - a. Double-click the Assignment Service shape to open the property pane.
    - a. In the **Assignment Service** field, enter a name for the shape.  
Choose a name that is meaningful to users who see the name on the work object history display, the breadcrumbs control (for entry points), and inside the Assignment Service shape on the **Diagram** tab.  
  
The task name does not affect run-time processing of the flow.
    - b. In the **Type** field, press the Down arrow key, and then select the name of a connect activity.
    - c. If the activity accepts input parameters, enter parameter values in the list that is displayed.
    - d. **Optional:** To associate a service-level agreement with the Assignment Service shape that is independent from service-level agreements set on a flow or case type, press the Down arrow key, and then select the name of a service-level agreement that you want to apply.
    - e. **Optional:** To control the text of instances that are added to the work item history when flow processing completes this shape, in the **Audit Note** field, enter or select the name of a *Rule-Message* rule.
    - f. **Optional:** To update the status of a case on this shape, in the **Work status** field, press the Down arrow key, and then select a value that you want to apply.  
For more information, see [Changing case statuses \(on page 318\)](#).
    - g. **Optional:** To activate a ticket in a flow that detects an exception, error flow, or event, in the **Tickets** section, in the **Ticket name** and the **Display name** fields, press the Down arrow key, and then provide values that you want to apply.

**Note:** You can use more tickets by adding a row for each ticket. If a shape has more than one ticket associated with it, then the processing of the task continues only after all tickets are set.

For more information, see [Responding to business exceptions in a flow \(on page 348\)](#).

3. Click **Submit**.

4. Click **Save**.

---

[Integrating a case with an external resource \(on page 374\)](#)

Activity form - how to create activities for flows (on page )

## Adjusting a service-level agreement for the Assignment Service shape

Meet your business goals by adjusting the service-level agreement defined for the Assignment Service shape in a process. By defining goals and deadlines, you can set expectations for the time that it takes an external system to complete a single task in the process.

**Note:** Service-level agreements that you define on shapes override service-level agreements that are set on processes and case types.

Service-level agreements on shapes override service-level agreements that are set on processes and case types.

1. Open the process that contains the Assignment Service shape that you want to modify:
  - a. In the navigation pane of Dev Studio, click **App**.
  - b. On the **Classes** tab, expand the case type for which you want to add the assignment, and then click **Process > Flow**.
  - c. In the list of process instances, double-click a process that you want to open.
2. Modify the SLA for the Assignment Service shape:

- a. Double-click the Assignment Service shape to open the property pane.
- b. In the **Service level** field, press the Down Arrow key, and then select the name of a service-level agreement.



**Note:** If you do not select an option, the system applies the *Work-.Default* service-level agreement.

3. Click **Submit**.
4. Click **Save**.


[Adjusting initial urgency for an assignment \(on page 332\)](#)

[Setting service-level agreements \(SLAs\) for stages, processes, and steps \(on page 330\)](#)

[Creating a service-level agreement \(SLA\) rule \(on page 332\)](#)

## Connecting a process to an external application

Use the Integrator shape in a process to send or receive data from an external application in a way that continues flow processing as soon as the Connect activity completes. By integrating external applications with your process, you can add more functionality to your application. For example, you can use the Integrator shape to retrieve customer account balances or verify account numbers in an external database.

The Integrator shape  identifies an activity that connects your instance of Pega Platform™ to an external system to send or receive data. The Integrator shape runs synchronously, meaning that it must receive a response immediately, and then moves a case to the next step.



**Tip:** To improve efficiency, as a best practice use data pages to retrieve information from an external system. You can use a data page with an autopopulated property or with an autopopulated property backed by a declare expression. For more information, see [Data page definition \(on page 332\)](#).

1. Add the Integrator shape to your process.
  - a. In the navigation pane of Dev Studio, click **App**.
  - b. On the **Classes** tab, expand the case type in which you want to add the assignment, and then click **Process > Flow**.

- c. In the list of Flow instances, double-click a flow that you want to open.
  - d. On the Flow form, on the **Diagram** tab, click the **Flow shapes** icon, and then click **Advanced Shapes > Integrator**.
  - e. Drag the Integrator shape to the flow diagram, and place the shape based on the order of events in the process.
  - f. Connect the Integrator shape by dragging connector end points to connection points on different shapes in the process.
2. Modify the Integrator shape to send a request from a process to an external application.
- a. Double-click the Integrator shape to open the property pane.
  - b. In the **Integrator** field, enter a name for the shape.  
Choose a name that is meaningful to users who see this on the work object history display, in the breadcrumbs control (for entry points), and inside the Integrator shape on the **Diagram** tab.  
  
The task name does not affect run-time processing of the flow.
  - c. In the **Rule** field, press the Down arrow key, and then select the name of a connect activity.
  - d. If the connector requires parameters, the system provides a list of response and request data transforms generated by the accelerator. Use the generated request data transform to define the mapping of application (source) properties to connector request (target) properties. Use the generated response data transform to define the mapping of connector response (source) properties to application (target) properties.
  - e. **Optional:** To control the text of instances added to the work item history when flow processing completes this shape, in the **Audit Note** field, select or enter the name of a *Rule-Message* rule.
  - f. **Optional:** To provide a link to this step from the breadcrumb trail navigation, select the **Enable navigation link** check box.
  - g. **Optional:** To update the status of a case on this shape, in the **Work status** field, press the Down arrow key, and then select a value that you want to apply.  
For more information, see [Changing case statuses \(on page 318\)](#).
  - h. **Optional:** To activate a ticket in a flow that detects an exception, error flow or event, in the **Tickets** section, in the **Ticket name** and the **Display name** fields, press the Down arrow key, and then select the values that you want to apply.

**Note:** You can use more tickets by adding a row for each ticket. If a shape has more than one ticket associated with it, then processing of that task continues only after all tickets are set.

For more information, see [Responding to business exceptions in a flow \(on page 348\)](#).

3. Click **Submit** to close the property pane.
4. Click **Save**.

---

[Creating a stand-alone process \(on page 249\)](#)

Activities (on page )

## Running a decision strategy from a process

You can use the Run Data Flow Decision shape to run a single case data flow that runs a decision strategy as part of the business process. By integrating decision strategies with your flow, you create decision logic that is contextual for each case.

1. Add the Run Data Flow shape to your process.
  - a. Open a process by searching for it or by using the Application Explorer. For more information, see [Finding rules by class \(on page \)](#).
  - b. On the **Diagram** tab, click the **Flow shapes** icon, and then click **Run Data Flow**.
  - c. Drag the Run Data Flow shape to a position on the flow diagram, based on the order of events in the flow.
  - d. Connect the Run Data Flow shape, by dragging connector end points to connection points on different shapes in the flow.
2. Double-click the Run Data Flow shape to open the property panel.
3. **Optional:** To use properties and other rules that are not in your current class path, change the context of the data flow.
  - a. In the **Define data flow on** list, select **Another page**.
  - b. In the **Page property** field, press the Down Arrow key, and then select a page property that defines pages of a class that is different from the current flow.
4. In the **Rule** field, press the Down Arrow key, and then select a data flow that references a strategy. The data flow must have an abstract input and abstract or dataset output.
5. **Optional:** For data flows that do not save output records, or have an abstract destination, press the Down Arrow key in the **Output page** field, and then select a page property that stores the results from the data flow.

6. Click **Submit** to close the property panel.
7. Click **Save**.

[Creating a stand-alone process \(on page 249\)](#)

[Types of data flows \(on page \)](#)

[About Strategy rules \(on page \)](#)

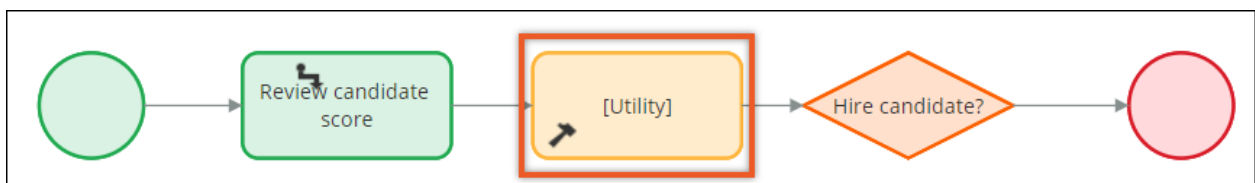
[Flow shapes \(on page 251\)](#)

## Calling an activity or automation from a process

Perform processing that does not require human interaction by calling an activity or automation from a process. As a result, you can create simpler, more efficient processes that have fewer shapes.

For example, an insurance company is required to upload submitted insurance claims to a government registry of motor vehicles. As part of the claim registration process, the application developer uses the Utility shape to call an activity or automation that uploads the submitted claim.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the **Life cycle** section, hover over a stage, and then click **Configure process**.
3. On the **Workflow** tab, click **Open process in Dev Studio**.
4. On the **Diagram** tab, click the **Flow shapes** icon, and then click **Utility**.
5. Drag the Utility shape to a position on the flow diagram, based on the order of events in the flow.
6. Connect the Utility shape by dragging connector end points to connection points on different shapes in the process.



Utility shape in a flow

7. Double-click the Utility shape to open the property pane.
8. In the **Select type of rule** field, select the type of rule.
9. Complete the following fields based on the type of rule that you are using:

Choices	Actions
<p><b>Run an activity rule</b></p> <p>For more information, see <a href="#">Creating an activity</a> (on page 374).</p>	<ol style="list-style-type: none"> <li>In the <b>Automation details</b> section, click <b>Activity</b>.</li> <li>In the <b>Rule</b> field, press the Down arrow key, and then select an activity of the Utility type.</li> <li>If the activity supports input parameters, in the additional fields, enter the values of the parameters that you want to set.</li> </ol>
<p><b>Run an automation rule</b></p> <p>For more information, see <a href="#">Automations</a> (on page 374).</p>	<ol style="list-style-type: none"> <li>In the <b>Automation details</b> section, click <b>Automation</b>.</li> <li>In the <b>Resource</b> field, select the type of automation that you want to use.</li> <li>In the <b>Name</b> field, select the automation to use.</li> <li>If the automation supports input parameters, in the additional fields, enter the values of the parameters that you want to set.</li> </ol>

10. Click **Submit**.

11. Click **Save**.

[Flow shapes](#) (on page 251)

[Integrating a case with an external resource](#) (on page 374)

<https://academy.pega.com/topic/activities/v3>

<https://academy.pega.com/topic/creating-activity/v2>

## Creating cases from a web mashup

Manage the business processes of users who use external software by embedding a case into their external web page in the form of a web mashup. For example, you can embed a case type for job application processing on a web page that displays job offers.





By generating mashup code in HTML and inserting it into an external web page or application, you enable users to interact with Pega features and create cases without having access to Pega Platform software. For more information, see Pega web mashups for embedding Pega Platform UI in external web pages (*on page* ).

**Note:** **Web embeds** replace web mashups in Cosmos React applications. For more information, see *Creating web embed (on page )* and *Embedding Pega Platform UI in web pages (on page )*.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Create new channel interface** section, click **Web mashup**.
4. On the **New Web mashup interface** form, in the **Basic options** section, in the **Name** field, enter a name for the mashup that you want to use to create case types.
5. **Optional:** To distinguish this channel interface from other Mashup channel interfaces, in the **Description** field, specify the purpose of this channel interface.
6. In the **URL** field, enter the URL for your mashup:
  - If you want the user to open the default application for their operator record, use the `https://sample.pegacorp.com/prweb` URL pattern.
  - If you want the user to access a specific application, use the `https://sample.pegacorp.com/prweb/app/<application alias>` URL pattern.
  - If your application uses an authentication service and you want users to open the default application for their Operator ID record, use the `https://sample.pegacorp.com/prweb/PRAuth/<authentication service alias>` URL pattern.
  - If your application uses an authentication service and you want users to access a specific application, use the `https://sample.pegacorp.com/prweb/PRAuth/app/<application alias>/<authentication service alias>` URL pattern.

**Note:**

- Because some users might have access to several applications in their Operator ID record, the best practice is to always use application aliases with mashups. For more information, see *Adding an application URL alias (on page )*.
- Mashups support all Pega authentication services only when the URL syntax in the `data-pegacorp-url` attribute is correct. For more information, see *Mashup attributes (on page )*.

7. **Optional:** To generate the mashup snippet with URL encryption, turn on the **Use encryption** switch.



**Note:** When the **Use encryption** switch is on, mashup requests do not support dynamic parameters.

8. In the **Configuration** section, in the **Action** list, select **Create a new case**.
9. In the **Case type** list, select a case type that you want to use.
10. In the **Flow name** field, enter the starting process of the case type .



**Note:** The starting process applies only to case types without the `Create` stage. For more information, see [The Create stage \(on page 48\)](#).

11. In the **Thread name** field, enter the name of the thread that handles the mashup.
12. **Optional:** To delay the load of the mashup gadget until a future point during page load or as part of a user interaction with the page, turn on the **Defer mashup load** switch.

You can then invoke the method to load the mashup at the appropriate time either within the onload event of the host page or after some other event on the page starts.

13. **Optional:** To retain the state of the case after users refresh the browser that displays the mashup, turn on the **Retain mashup state on browser refresh** switch.



**Note:** Consider the following limitations when using this option:

- The option to retain the case state is not backwards compatible by default.
- The option to retain the case state works only if users load the mashup gadget on the same thread, and the thread is available during a browser refresh. This setting does not work if users set the thread name dynamically while loading the mashup.
- The state is retained if the reload occurs in the same session.
- The option to retain the case state does not work for temporary work objects, for which the system does not generate work IDs.

14. **Optional:** To validate dynamic parameters in the mashup request, turn on the **Allow passing dynamic parameters** switch.
15. Configure the output options for the mashup:

- a. In the **Skin** list, select the skin rule that you want to use for the mashup.
- b. In the **Iframe resizing** list, select the resize mode for the mashup.
- c. In the **Initial skeleton** list, select the skeleton to provide a template for the mashup.



**Note:** You can check how the skeleton renders the content by clicking **Skeleton preview**.

16. **Optional:** To add a parameter, such as a parameter for custom authentication, in the **Custom parameters** section, click the **Add a row** icon, and then enter a parameter name and value.
17. In the **What is a mashup** section, click **Generate mashup code**, and then, in the **Mashup code** window, select the type of code that you want to copy:
  - To copy the iframe code, above the **iframe code** window, click **Copy**.
  - To copy the mashup code, above the **Mashup code** window, click **Copy**.



**Note:** The mashup code option also creates an iframe when the mashup loads. Selecting the iframe code improves the response time of a mashup but does not support passing dynamic parameters.

18. Close the **Mashup code** window.
19. Click **Save**.
20. **Optional:** To preview how different devices display the mashup code, in App Studio, click **Preview application**, and then select the channel that you want to preview by clicking the channel name next to the application name.

### Related information

[Case life cycle elements \(on page 43\)](#)

[Views for cases \(on page 168\)](#)

[Configuring case creation \(on page 223\)](#)

### Directed Web Access in configuring assignments for external users

When you enable the Directed Web Access (DWA) in your application, anyone who accesses the Internet, an intranet, or email can process an assignment. When you use this feature, you extend the reach of your application to employees throughout the enterprise, trusted customers and suppliers, and anyone else



from whom you want to obtain information. For example, you can configure your application to send an email with a DWA link to someone who has complained, which they can click, and then answer a feedback question on whether they are satisfied with the resolution of a complaint. Users might not be customers, so they might not have login credentials to access the website and application.

Assignments define the parts of case processing that require human judgment, expertise, and data entry from trained users of the application. Assignments that you send to outside parties are external assignments. External assignments improve accountability, eliminate the need for phone calls, and give visibility to the responsiveness of the parties.

To send an external assignment to users, the application includes a specially formatted URL in the text of an email. When the email recipient clicks the URL, a web browser session opens and submits a once-only identifier and password to the Pega Platform server.

After authenticating these values, Pega Platform sends an assignment to the user's browser. When the user completes and submits the assignment, the requestor connection ends. The external user cannot repeat the assignment or reuse the URL or password because the session is authenticated for one time only.

For security reasons, the URL for an external assignment must be static, which means that it cannot be generated or altered by JavaScript or other processing at run time. Because of this restriction, the flow action cannot use AJAX, dynamic select, or SmartPrompt, which require multiple server interactions.

External assignments are instances of the *Assign-External* class, or of its subclass. The standard activity *Work-External* creates these assignments.

The following run-time events describe the way that users interact with external assignments:

1. A case reaches an external assignment in the life cycle.
2. Your application sends an email to the user who is identified in the routing settings for the external assignment.
3. The user clicks the URL in the email message, which sends a token to your application for authentication.
4. The web browser of the user displays the form for the assignment.
5. The user completes the assignment by entering values in the fields on the form and submitting the form.
6. The session ends and the case moves to the next step in the life cycle.
7. Your application updates the count of web invocations that are used in reports for license compliance.

---

[Sending automatic emails from cases \(on page 142\)](#)

[Requesting approval by email and push notification \(on page 93\)](#)


## Configuring Directed Web Access

Configure the Directed Web Access (DWA) capability to support external assignments in a case. By collecting information from stakeholders who are outside of your organization, you can extend the scope of your application and provide necessary details for case processing. For example, in a mortgage loan application, for the house valuation stage, you need a note from an external agency that does the valuation of the house. To request the note, send an email link, and then attach the received note to the loan case.

1. Update the operator ID for the external user:
  - a. In the header of Dev Studio, click **Configure > Org & Security > Organization > Operators**.
  - b. In the list of operators, search for and open the **External** operator ID.
  - c. On the **Profile** tab of the operator rule form, in the **Access Group** section, add a new access group item that gives external users access to your application rulesets.
  - d. On the **Work** tab, in the **Team** field, add the work group to which the user belongs.
  - e. On the **Security** tab, ensure that the **License type** field is set to **Invocation** and that the **Use external authentication** and **Disable Operator** check boxes are empty.
  - f. Click **Save**.
2. Update the Public Link URL:
  - a. In the header of Dev Studio, click **Configure > System > Settings > URLs**.
  - b. Complete the **Public Link URL** field with the URL that you want to convey to external users.
3. Ensure that you have a valid email account and that the system is configured to send and receive email:
  - a. In the header of Dev Studio, click **Configure > Integration > Email > Email Accounts**.
  - b. Create an email account in the appropriate class group.

For more information, see [Configuring outbound email in App Studio \(on page 142\)](#).
  - c. Click the **Test connectivity** link and ensure that you have the outbound email rule for the application running.
4. Configure a correspondence rule that contains the DWA URL:

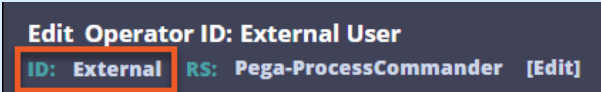


- a. Create a correspondence rule of the *Rule-Obj-Corr* type.  
For more information about how to open a create rule form, see [Creating rules \(on page 389\)](#).
- b. In the correspondence rule form, in the **Label** field, enter a name that starts with `External`.
- c. In the **Correspondence Type** field, select **Email**.
- d. Click **Create and Open**.
- e. In the correspondence rule form, add an external correspondence fragment, such as *ExternalInstructions*, *ExternalDWAInstructions*, or *workLink*, by clicking the **Insert Rule**  button, and then click **Save**.

5. Add the external user as a work party for your case type:

- a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
- b. On the **Settings** tab, click **Participants**.
- c. In the **Participant configuration** form, in the **Role name** field, enter the operator ID for the external user.

**Note:** Ensure that you use the operator ID, not the operator description:



For more information about how to configure a case participant, see [Configuring case participants \(on page 240\)](#).

- d. In the **Data transform** field, enter a name for a new data transform.
- e. In the data transform configuration form, on the **Definition** tab, in the **Target** field, enter `.pyWorkPartyUri`.  
Each party represented in a work object has a unique identifier, recorded in the *pyWorkPartyUri* standard property. Depending on the class or other processing, it can be an operator ID, email address, account number, or other information.

- f. In the **Source** field, enter the operator ID for the external user.
  - g. Click **Save**.
  - h. Select the **Create participant automatically when the case starts** checkbox.
  - i. Click **Done**.
  - j. Click **Save**.
6. Configure the assignment shape:
- a. In the navigation pane of Dev Studio, click **App**.
  - b. On the **Classes** tab, expand the case type in which you want to edit the assignment, and then click **Process > Flow**.
  - c. In the **Flow** form, on the **Diagram** tab, double-click an assignment.
  - d. In the **Assignment properties** dialog box, in the **Routing** section, from the **Route to** list, select **Custom**.
  - e. In the **Assignment type** list, select **External**.
  - f. In the **Router** field, select **ToWorkbasket**, and then provide the name of the work queue to which you want to route the assignment.
  - g. In the **Advanced** section, click the **Assignment details** node, and then, in the **OperatorModel** field, select the operator ID for the external user.
  - h. In the **DaysToExpiration** field, enter the number of days after which the DWA link expires and the external user can no longer open the assignment.
  - i. In the **Party** field, enter the operator ID, for example `External`.
  - j. In the **Subject** field, enter the email message subject.
  - k. In the **HarnessPurpose** field, select **PerformExternal**.
  - l. Click **Submit**.
  - m. Click **Save**.

At run time, the system identifies the party that is also an email recipient to receive the email correspondence, and then enters a subject line and message. The system creates the external assignment. The flow pauses until the external party completes the external assignment. The link in the external assignment does not require login, but the user can use it only once.

### Related information

- Operators (on page )
- Creating an operator ID (on page )
- Correspondence rules and email (on page )
- Correspondence rules (on page )



[About Flow Actions \(on page 302\)](#)

[Flow shapes \(on page 251\)](#)

## Enabling cases for creation through web and chat channels

To reach wider audiences, ensure that users of your application can create cases also outside Pega Platform by enabling cases for creation through web and chat channels. As a result, you help your customers reach their business goals in various scenarios that include different channels and flexibility of processing. For example, when you enable a Book a flight case type for a creation through a web channel, a customer can start a new case to buy a flight ticket directly through a website or a chat instead of logging to an application.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. Click the **Settings** tab, and then open the **General** section.
3. In the **Availability** section, select the **Show in 'New' menu** check box.
4. Click **Save**.

---

### Related information

[Creating cases from a web mashup \(on page 384\)](#)

## Sourcing attachments from external storage

Gain access to third-party documents and enhance collaboration between stakeholders by configuring external storage as a source of content that you can attach to your case, space, or Pulse conversation.

Sourcing from external storage enables you to use more relevant data in your business process. For example, you can configure an external source for attachments for CSRs who manage loan requests to ensure that they use the most up-to-date version of the necessary documents, such as the Terms and Conditions document.

As a best practice, use repositories for case attachments. Repositories use data streaming to present attachment data directly to your application. When you use web storage or CMIS, Pega Platform™ places data on the clipboard before use in the application, which uses more system resources.

Your environment type determines the repositories that you can configure for your external content for case, space, or Pulse attachments:






- In Pega Cloud environments, you can use `pegacloudfilestorage` and the `/attachments` subfolder as the repository for case attachments, or you can configure another type of repository.

Before integrating the repository in your application, you can change the repository that you use for case attachments to another external repository type after you create the repository.


For more information, see *Creating a repository (on page )*.

- In on-premise and client-managed cloud systems, you specify an external repository for case attachments after you create the repository.

For more information, see *Creating a repository (on page )*.

 **Important:** Do not use the internal Pega Platform database as a repository.

1. In the header of Dev Studio, click the name of the application, and then click **Definition**.
2. On the Application form, click the **Integration** tab.
3. In the **Content sourcing** section, configure at least one external storage system:

Choices	Actions
<p><b>A web service provider</b></p>	<ol style="list-style-type: none"> <li>a. Select the <b>Source from web storage providers</b> checkbox.</li> <li>b. In the <b>Provider</b> list, select an installed web storage provider for which you have an account and a content location.</li> <li>c. In the <b>Authentication profile</b> box, enter or select an authentication profile of the OAuth 2.0 type.</li> </ol> <div data-bbox="938 1520 1464 1751" style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p> <b>Note:</b> Use the same authentication profile to store content in and source from the web storage provider.</p> </div> <ol style="list-style-type: none"> <li>d. <b>Optional:</b> To change the label that users see when they add an attachment</li> </ol>

Choices	Actions
<p data-bbox="240 541 412 575"><b>A repository</b></p>	<p data-bbox="938 226 1450 302">from this web storage provider, enter a description in the <b>Label</b> field.</p> <p data-bbox="906 344 1425 512">e. <b>Optional:</b> To use content from more than one web storage provider, click <b>Add</b>, and then repeat these steps to configure another provider.</p> <p data-bbox="906 541 1450 785">a. Select the <b>Source from repositories</b> checkbox.</p> <p data-bbox="906 659 1450 785">b. In the <b>Repository</b> list, select a repository for which you have an account and a content location.</p> <div data-bbox="938 848 1461 1079" style="border: 1px solid #0070C0; padding: 5px; margin: 10px 0;"> <p data-bbox="948 945 984 987">i</p> <p data-bbox="1019 877 1445 1058"><b>Note:</b> Pega Cloud clients: To use Pega Cloud File storage, select the <code>pegacloudfilestorage</code> repository.</p> </div> <p data-bbox="906 1150 1458 1226">c. Click <b>Browse</b>, and then select the folder from which to source attachments.</p> <div data-bbox="938 1289 1461 1520" style="border: 1px solid #0070C0; padding: 5px; margin: 10px 0;"> <p data-bbox="948 1386 984 1428">i</p> <p data-bbox="1019 1318 1432 1499"><b>Note:</b> Pega Cloud® clients: To use Pega Cloud File storage, select the <code>/attachments</code> subfolder.</p> </div> <p data-bbox="906 1591 1094 1625">d. Click <b>Select</b>.</p> <p data-bbox="906 1667 1458 1743">e. <b>Optional:</b> To change the label that users see when they add an attachment</p>

Choices	Actions
<p><b>CMIS systems</b></p>	<p>from this repository, in the <b>Label</b> field, enter a description.</p> <p>f. <b>Optional:</b> To use content from more than one folder, click <b>Add</b>.</p> <hr/> <p>a. Select the <b>Source from CMIS systems</b> checkbox.</p> <p>b. <b>Optional:</b> To change the label that users see when they add an attachment from this source, in the <b>Label for this content source (where available)</b> field enter a description.</p>

4. Click **Save**.

Your content from external storage becomes available when you add an attachment to a case, space, or Pulse conversation.

File and content storage (on page )

[Storing case attachments using external storage \(on page 400\)](#)

[Attaching content to a case \(on page 101\)](#)

[Attachment types \(on page 458\)](#)

Creating an authentication profile (on page )

Integrating your application with external resources (on page )

## Adding attachments from external locations

Provide more data and relevant context to your work by supplementing your case, space, or Pulse conversation with external content.

Log in to a portal, and then add an attachment from an external location:

## Choices

### Add a case attachment

## Actions

1. Create or find a case to which you want to attach external content.



**Note:** You can locate a case by searching for it or by looking in your worklist or the **Recents** list.

For more information, see [#unique\\_270 \(on page \)](#).

2. In the upper-right corner, click the **Expand the utility pane** icon.
3. In the **Files & documents** section, click the **Manage files and documents** icon, and then in the **Manage content** dialog box, click the menu option from which you can add files from an external system.

The options in the menu show the values that you provide in the **Label** field on the **Integration** tab of your application rule. For more information, see [Sourcing attachments from external storage \(on page 392\)](#).



**Note:** When you connect to your external storage account from the application for the first time, you might receive a prompt to authenticate and authorize the connection.

4. Attach a file from your external system.

Choices	Actions
	<p data-bbox="901 247 1446 478"><b>Note:</b> Attaching multiple files with the same name to the same case results in an error because storage providers treat the file name together with the path as a key.</p> <p data-bbox="865 562 1073 598">5. Click <b>Submit</b>.</p> <p data-bbox="980 632 1446 709">The case attachment is displayed in the <b>Files &amp; documents</b> section.</p> <p data-bbox="980 806 1419 982"><b>Note:</b> To remove a case attachment, click the <b>Delete</b> icon next to the name of the attachment.</p>
<b>Add a space attachment</b>	<ol data-bbox="865 1262 1458 1831" style="list-style-type: none"><li>1. In the navigation pane, click <b>Spaces</b>.</li><li>2. Click the space to which you want to add content.</li><li>3. In the upper-right corner, click the <b>Expand the utility pane</b> icon, and then click the <b>Related</b> tab.</li><li>4. In the <b>Files &amp; documents</b> section, click the <b>Manage files and documents</b> icon, and then in the <b>Manage content</b> dialog box, click the menu option from which you can add files from an external system.</li></ol>

## Choices

## Actions

The options in the menu show the values that you provide in the **Label** field on the **Integration** tab of your application rule. For more information, see [Sourcing attachments from external storage \(on page 392\)](#).

- Attach a file from your external system.



**Note:** Attaching multiple files with the same name to the same case results in an error because storage providers treat the file name together with the path as a key.

- Click **Submit**.

The space attachment is displayed in the **Files & documents** section.



**Note:** To remove a space attachment, click the **Delete** icon next to the name of the attachment.

## Add a Pulse attachment

- Find or create a Pulse message to which you want to add content. For more information, see [Viewing your activity feed \(on page 507\)](#) and [Posting a message in Pulse \(on page 499\)](#).

Choices	Actions
	<p>2. In the field that is displayed in the <b>Pulse</b> section, click <b>Add attachments &gt; External Sources</b>, and then click an option from which you can add files from an external system.</p> <p>The options in the menu show the values that you provide in the <b>Label</b> field on the <b>Integration</b> tab of your application rule. For more information, see <a href="#">Sourcing attachments from external storage (on page 392)</a>.</p> <p>3. Attach a file from your external system.</p> <p>4. Click <b>Post</b>.</p> <div data-bbox="899 898 1461 1016" style="background-color: #e1f5fe; padding: 5px; border: 1px solid #0070c0;"> <p>The Pulse attachment is displayed in the <b>Pulse</b> section.</p> </div>

## Adding case attachments from a CMIS system

Enable users to access content stored in external sources by adding an attachment from a CMIS system to a case. Downloading content from an external CMIS system guarantees that users use the most up-to-date version of documents.

Content Management Interoperability Services (CMIS) are external systems that use the CMIS standard, such as Alfresco, Microsoft SharePoint, and OpenText, to which you connect by using a CMIS connect rule. CMIS systems support storing and sourcing content. For more information, see [About Connect CMIS rules \(on page 392\)](#).

You can use any CMIS-compatible server to source data for your application.

1. Log in to a portal, and then create or find a case or space to which you want to attach a file from a CMIS system.
2. In the upper-right corner, click the **Expand the utility pane** icon.
3. In the **Files & documents** section, click the **Manage files and documents** icon, and then in the **Manage content** dialog box, click the **Content from CMIS** tab.
4. In the **Select a file** section, select a file from one or more CMIS systems.



- The CMIS systems in the list correspond to the *Connect-CMIS* rules that are available for the current operator.
  - The files in a CMIS system come from the CMIS repository that you configure in your *Connect-CMIS* rule that references the system.
5. In the **Name** field, provide a name for the attachment.
  6. Click **Submit**.

The attachment is displayed in the **Files & documents** section.

7. **Optional:** To remove a case attachment, click the **Show more** icon next to the name of the attachment, and then click **Delete**.

## Managing case attachments stored in a CMIS system

Manage versions of a case attachment that is stored in a CMIS system at run time to control which version of the attachment you want users to download.

1. Click the **Gear** icon next to the name of the attachment.
2. In the **Manage attachment** window, in the **Version history** section, click the file name for the appropriate version of the attachment to download.
3. **Optional:** To prevent other users from modifying the file attachment, select **Lock this attachment for local editing**.
4. Click **Upload new version** to check in the new version of the file.

The new version is displayed in the **Version history** section.

### Storing case attachments using external storage

Upload large files without causing out-of-memory errors by configuring your application to store case and Pulse attachments outside the Pega Platform™ database. For example, you can configure an external storage for attachments for insurance agents to ensure that the system always has enough memory even with heavy upload of files that support insurance claims.

The default storage for on-premises deployments is the Pega database. As a best practice, use repositories for case attachments. Repositories are the only supported content storage type that supports streaming and that has no risk of running out of memory.

Your environment type determines the repositories that you can configure for your external content for case or Pulse attachments:






- In Pega Cloud environments, you can use `pegacloudfilestorage` and the `/attachments` subfolder as the repository for case attachments, or you can configure another type of repository.

Before integrating the repository in your application, you can change the repository that you use for case attachments to another external repository type after you create the repository.

For more information, see [Creating a repository \(on page 398\)](#).

- In on-premises and client-managed cloud systems, you specify an external repository for case attachments after you create the repository.

For more information, see [Creating a repository \(on page 398\)](#).

 **Important:** Do not use the internal Pega Platform database as a repository.

For more information about different types of content storage types that you can use, see [Integrating with file and content management systems \(on page 398\)](#) and [File and content storage \(on page 398\)](#).

**Note:** To access the contents of a repository that stores attachments, you can use the `D_pxGetFile` API.





For more information, see [Repository APIs \(on page 398\)](#) and [Using repository APIs in your application \(on page 398\)](#).

1. In the header of Dev Studio, click the name of the application, and then click **Definition**.
2. On the Application form, click the **Integration** tab.
3. In the **Content storage** section, configure at least one external storage system:

Choices	Actions
CMIS systems	<ol style="list-style-type: none"> <li>a. Select the <b>Store in CMIS system</b> checkbox.</li> <li>b. In the <b>Connector name</b> field, select a connector.</li> </ol>

Choices	Actions
<p><b>A web service provider</b></p>	<ul style="list-style-type: none"> <li>a. Select the <b>Store in web storage provider</b> checkbox.</li> <li>b. In the <b>Provider</b> list, select an installed web storage provider for which you have an account and a content location.</li> <li>c. In the <b>Authentication profile</b> box, enter or select an authentication profile of the OAuth 2.0 type.</li> </ul> <div data-bbox="938 705 1463 936" style="border: 1px solid #0070C0; background-color: #D9E1F2; padding: 10px; margin: 10px 0;"> <p><b>Note:</b> Use the same authentication profile to store content in and source from the web storage provider.</p> </div> <ul style="list-style-type: none"> <li>d. Click <b>Browse</b>, and then select the target folder for your attachments.</li> <li>e. Click <b>Select</b>.</li> <li>f. <b>Optional:</b> To display a prompt to users before they delete an attachment, click the <b>Prompt users for confirmation before deleting content from storage location</b> checkbox.</li> </ul>
<p><b>A repository</b></p>	<ul style="list-style-type: none"> <li>a. Select the <b>Store in repository</b> checkbox.</li> <li>b. In the <b>Repository</b> list, select a repository for which you have an account and a content location.</li> </ul> <div data-bbox="938 1745 1463 1854" style="border: 1px solid #0070C0; background-color: #D9E1F2; padding: 10px; margin: 10px 0;"> <p><b>Note:</b> Pega Cloud clients: To use Pega Cloud File storage, select</p> </div>

Choices	Actions
	<p data-bbox="941 220 1468 336">  the <code>pegacloudfilestorage</code> repository.         </p> <p data-bbox="901 399 1468 483">c. Click <b>Browse</b>, and then select the folder for your attachments.</p> <p data-bbox="941 546 1468 777">  <b>Note:</b> Pega Cloud® clients: To use Pega Cloud File storage, select the <code>/attachments</code> sub-folder.         </p> <p data-bbox="901 840 1468 882">d. Click <b>Select</b>.</p> <p data-bbox="901 913 1468 1140">e. <b>Optional:</b> To display a prompt to users before they delete an attachment, click the <b>Prompt users for confirmation before deleting content from repository</b> checkbox.</p>

4. Click **Save**.

Pega Platform adds new attachments that you create in a case or Pulse conversation to the location that you choose.

Integrating with file and content management systems (on page )

File and content storage (on page )

## Limitations for attachments in a file storage repository

When you configure your application to store case or Pulse attachments in a repository, consider the limitations to ensure that your attachments are stored and retrieved correctly.

Ensure that you review the following limitations for adding a repository for case attachments:

- Repositories support only file attachments. All other types of attachments reside in a database.
- You cannot change the content storage type from repository to a different type.



- After you switch from database storage to repository storage, you cannot switch back to database storage without losing access to the attachments in the repository.
- You cannot modify the configuration of a repository after the repository is in use.
- Application attachments that you provide on the application form reside in a database.
- Multiple applications in the same application rule stack must have the same repository configuration, or users cannot access the file attachments.
- Because the application Pulse gadget cannot prompt for authentication details, Pulse attachments that you provide on the Application Overview landing page reside in the repository only when your session is already authenticated or when the repository supports non-interactive authentication. If both of these conditions are false, an error occurs and the operation fails.
- The following features are not supported for custom repositories during DevOps:
  - Import and export
  - File listener
- For large images and PDF attachments, poster creation is skipped, and then Pega Platform™ creates a placeholder thumbnail.
- When using database storage, you can attach multiple files that have the same name to the same case. For external storage, you cannot attach multiple files that have the same name to the same case because storage providers treat the file name plus the path as a key. Attempting to attach multiple files that have the same name to a case causes an error. You can, however, attach files that have the same name to different cases.

For more information, see [Creating a repository](#) (on page [100](#)) and [File and content storage](#) (on page [100](#)).

**Note:** When you configure your application to store case attachments in a repository, Pega Platform appends the case ID to the attachment name. For example, an attachment called `loanapplication.pdf` becomes `loanapplication_CASE-1234.pdf`.

- The unsupported characters that might not work in attachment names are {""", "?", "\*", "<", ">", "|", ":", "}".

## Migrating case attachments to a repository

Reduce the size of your Pega Platform™ database and improve performance by migrating the case attachments for your application from a Pega Platform database storage to a repository.





**Note:** When you migrate your applications from on-premises to Pega Cloud® deployments, move your attachments from a Pega database to a repository.

1. In the header of Dev Studio, click **Configure > System > Release > Upgrade > Attachment Migration**.
2. In the **Repository** field, select the repository to which you want to migrate case attachments.
3. Click **Browse**, navigate to the folder on your repository that you want to use for case attachments, and then click **Select**.
4. Click **Start Migration**.
5. Click **Refresh** to update the display.

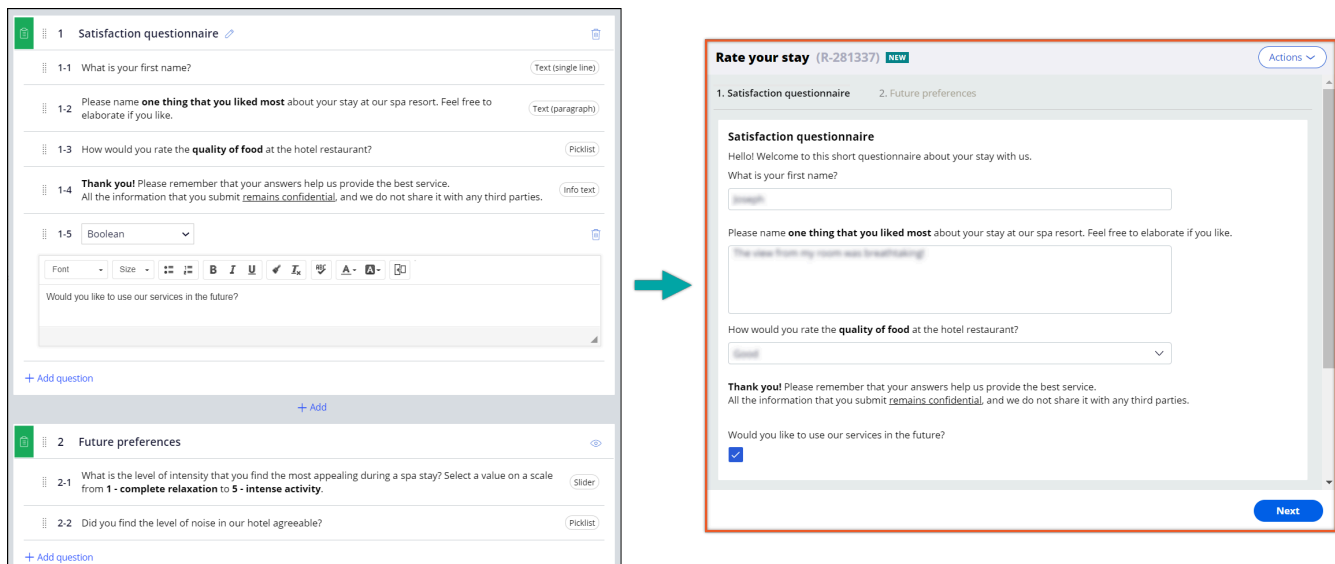
After the migration finishes, the results include the number of migrated files. For the files that were not migrated, the system work object ID and a description of the error, which you can use for troubleshooting.

## Designing questionnaires

Gather information quickly from users while they process a case by designing a questionnaire. When users complete a questionnaire, the answers that they provide can capture the details that you need to successfully resolve your Microjourney.

By taking advantage of the multiple formats that questions can take in your questionnaires, you ensure that you capture the precise information that your business processes require. For example, you can provide a text box in which users can enter their answers directly, or create a radio button matrix, so that users can select their answers from a group of radio buttons. To save time, you can collect related questions on one page, and then reuse that page in other questionnaires. For instance, you can create a question page that captures details about customer satisfaction, and then reuse that question page in questionnaires about different business processes.

The following figure shows the process of authoring a questionnaire that collects a customer's answers about their stay at a spa resort, and the same questionnaire at run time:



Designing a questionnaire and running it in your application

### Running a questionnaire in a case (on page 138)

## Creating a questionnaire

Collect a wide range of information from users by providing questionnaires in a low-code way in App Studio. By including questions in different formats, you ensure that you gather the precise data that you need for your business processes. For example, after a customer service representative (CSR) finishes processing an insurance claim review case for a customer, the customer can complete a questionnaire to provide feedback on the CSR's performance.

Questionnaires store answers to questions in data objects in a data model. You can reuse data objects across your application. For example, if a customer provides a case ID in a questionnaire, you can use the data object that stores the case ID in other, related cases. To speed up the creation of your questionnaire, you can then upload an **.xlsx** or **.csv** file that contains a data model.

1. In the navigation pane of App Studio, click **Case types**.
2. In the header of the **Case types** landing page, click **New**.
3. In the **Create case type** dialog box, in the **Case type name** field, enter the name of your questionnaire.
4. Expand the **Advanced** section, and then, in the **Type** section, select **Questionnaire**.
5. **Optional:** To speed up the development of your questionnaire, reuse assets from another case type:
  - a. Expand the **Advanced** section.
  - b. In the **Reuse assets from** list, select a case type that you want to reuse.
6. **Optional:** To provide fields for your questionnaire upon creation, upload a file that contains a data model:

- a. Expand the **Advanced** section, and then select the **Create data model using spreadsheet** check box.
  - b. Click **Next**.
  - c. In the **Select file** dialog box, click **Choose File**.
  - d. Navigate to the file with the data model, and then click **Open**.
  - e. Click **Next**.
  - f. In the **Create fields** dialog box, specify field names and field types, and then click **Create**.
7. Click **Save**.

[Configuring a data model for a case \(on page 150\)](#)

[Running a questionnaire in a case \(on page 138\)](#)

[Best practices for creating questionnaires \(on page 423\)](#)

## Adding a question page to a questionnaire

To organize related questions, create a question page. You can reuse a question page across multiple questionnaires, and as a result, speed up the development of your application. For example, you can create a question page to store questions about the service at a hotel. Then you can reuse the question page for questionnaires that gather answers about the quality of the hotel restaurant, and the room service. When customers complete the questionnaires, they interact with different questions about the restaurant and the hotel, but the questions about the service quality are the same.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. On the **Questionnaire** tab, add a question page:

- To create a new question page, click **Add questionnaire item > Question page**.
  - To reuse an existing question page, click **Add questionnaire item > Library question page**, and then, in the **Questionnaire library** dialog box select a question page to add, and click **Add**.
3. In the **Title** field, click **Click here to edit page title**, and then enter a unique name for the question page, for example *Service quality*.
  4. **Optional:** To display the question page only in specific scenarios, define visibility conditions. For more information, see [Conditionally displaying a question page in a questionnaire \(on page 428\)](#).
  5. **Optional:** To provide additional guidance for users, add instructions that an application displays at the top of the question page:
    - a. In the **Page settings** pane, in the **Instructions** section, click the **Properties** icon.
    - b. In the rich text editor, enter your text.
    - c. Click **OK**.
  6. **Optional:** To manage your question pages in an organized way, categorize the question page:
    - a. In the **Page settings** pane, in the **Category** section, click the **Properties** icon.
    - b. In the **Category** field, enter the category that you want to associate with the question page, and then click **OK**.
  7. Click **Save**.

---

[Conditionally displaying a question page in a questionnaire \(on page 428\)](#)

[Running a questionnaire in a case \(on page 138\)](#)

## Adding a question page to a survey

Add a question page to a survey to organize related questions.

1. In the header of Dev Studio, click **Configure > Case Management > Questionnaire**.
2. Click the name of a survey.
3. Click **+ Add survey item > New question page**.



**Tip:** To save time, reuse a question by page by clicking **+ Add survey item > Library question page**.

4. In the **Page title** field, click the **Edit** icon, and then enter a unique name for the question page.
5. Click **Save survey**.



## Conditionally displaying a question page in a questionnaire (on page 428)

### Adding a question to a question page

Prompt users for information in a format that you define by adding a question to a question page. At run time, users interact with questions that you provide and deliver data that your business processes require to reach resolution. By engaging multiple types of questions, you can collect the exact information that you need. For example, you can create a text box question and ask users to provide their name.

For improved management of data in your application, you associate the questions with fields from a data model. The fields store the answers to the questions that users provide at run time. To meet your precise business needs, you can mark questions as required to ensure that users enter these answers before submitting the questionnaire. You can also validate the answers to avoid issues and help users provide information in an expected format.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <b>Questionnaire</b> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. On the **Questionnaire** tab, on the question page that you want to edit, add a question:

Choices	Actions
<b>Create a new question</b>	<ol style="list-style-type: none"> <li>a. Click <b>Add question &gt; New</b>.</li> <li>b. In the list of question types, select a type that you want to use, for example, select <b>Text (paragraph)</b>.</li> <li>c. In the text editor, enter the text of the question.</li> </ol>

Choices	Actions
	<p>d. If a question type that you want to create requires additional formatting, provide relevant configurations.</p> <p>For more information about formatting questions, see:</p> <ul style="list-style-type: none"><li>• <a href="#">Formatting text boxes in a questionnaire (on page 412).</a></li><li>• <a href="#">Formatting a picklist in a questionnaire (on page 413).</a></li><li>• <a href="#">Formatting a checklist in a questionnaire (on page 415).</a></li><li>• <a href="#">Formatting a slider in a questionnaire (on page 416).</a></li><li>• <a href="#">Formatting a radio button matrix in a questionnaire (on page 417).</a></li><li>• <a href="#">Formatting a currency question in a questionnaire (on page 419).</a></li><li>• <a href="#">Formatting a decimal question in a questionnaire (on page 421).</a></li></ul> <div data-bbox="938 1136 1468 1799" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 10px;"><p><b>Note:</b> Only standard Pega Platform applications support most questions that require formatting. Applications based on Cosmos React do not display formatted questions. A picklist is the only formatted question that you can use both in standard Pega Platform and Cosmos React applications. You can use currency and decimal questions only in Cosmos React applications.</p></div>

Choices	Actions
<b>Reuse an existing question</b>	<ol style="list-style-type: none"> <li>Click <b>Add question &gt; From library</b>.</li> <li>In the <b>Questionnaire library</b> dialog box, select a question that you want to reuse, and then click <b>Add</b> next to the question.</li> <li>Close the dialog box by clicking <b>Add</b>.</li> </ol>

3. In the **Map answer to field** list, select a field in the data model that stores the answer:

Choices	Actions
<b>Reuse an existing field</b>	In the list, select the field from a data model.
<b>Create a new field</b>	<ol style="list-style-type: none"> <li>In the list, click <b>Create New</b>.</li> <li>In the <b>Add field</b> dialog box, in the <b>Field name</b> field, enter the name, for example, enter <code>SatisfactionLevel</code>. The field type is the same as the question type.</li> <li><b>Optional:</b> To provide additional configuration for the field, expand the <b>Advanced</b> section, and then provide a new field ID, a description, and the maximum number of characters of the value that the field stores.</li> <li>Click <b>Submit</b>.</li> </ol>

4. **Optional:** To ensure that users answer a question, mark the question as required.

For more information, see [Making a question required in a questionnaire \(on page 426\)](#).

5. **Optional:** To ensure that users provide answers in an expected format, create a validation.

For more information, see [Validating answers in a questionnaire \(on page 431\)](#).

If the validation fails at run time, users receive an error message. Users need to correct the answers with failed validation before submitting the questionnaire.

6. **Optional:** To display the question only in selected scenarios, configure visibility conditions.

For more information, see [Conditionally displaying a question in a questionnaire \(on page 427\)](#).

At run time, the questionnaire displays the question only if the visibility conditions evaluate to true.

7. **Optional:** To manage your questions efficiently, categorize the question:
  - a. In the **Question settings** pane, in the **Category** section, click the **Properties** icon.
  - b. In the **Category** field, enter the category, for example, `CustomerService`.
  - c. Click **OK**.
8. **Optional:** To add more questions to the question page, repeat steps 2 (on page 409) through 7 (on page 412).
9. Click **Save**.

[Creating questions in bulk \(on page 444\)](#)

[Creating a stand-alone question with a single answer \(on page 436\)](#)

[Defining conditions in the condition builder \(on page \)](#)

[Running a questionnaire in a case \(on page 138\)](#)

## Formatting text boxes in a questionnaire

Format multiple text boxes in a questionnaire to ask users to provide short text answers. For example, in a questionnaire for job candidates, you ask users to provide their university major, previous employer, and a name of their own business if they have one.



**Note:** Only standard Pega Platform applications support questions with multiple text boxes. Applications based on Cosmos React do not display questions with multiple text boxes.

1. Navigate to the questionnaire:

Choices	Actions
<p><b>Open the questionnaire in App Studio</b></p>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>

Choices	Actions
<b>Open the questionnaire in Dev Studio</b>	<ul style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ul>

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Multi-text boxes**.
4. In the text field, enter the text of the question.
5. Label the text boxes:
  - a. In the **Textbox options** section, click **Add value**.
  - b. In the text field, enter a label for the text box.
6. **Optional:** To display the text box only in selected situations, define conditions:
  - a. Click **More > Advanced view**.
  - b. From the **Visible when** list, select **When**, and then click **Edit**.
  - c. Provide a value to compare at run time, a comparator, and a value to compare against the first value.
  - d. **Optional:** To add more conditions, click **Add a row**.
  - e. If you create multiple rows, define whether an answer needs to meet all or any conditions by selecting either **and** or **or** between the rows.
  - f. Click **Submit**.
7. Click **Save**.

---

[Designing questionnaires \(on page 405\)](#)

[Defining conditions in the condition builder \(on page \)](#)

[Fine-tuning your questionnaire \(on page 426\)](#)

## Formatting a picklist in a questionnaire

Format a picklist in a questionnaire to define the choices that users can select to answer a question. With a picklist, users can select only one answer to a question. For example, you can ask users about their rating of customer service in your company, and then provide a choice of answers that range from excellent to poor.



1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <b>Questionnaire</b> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Picklist**.
4. In the **Question settings** pane, in the **Map answer to field** list, select a field of the picklist type:

Choices	Actions
<b>Reuse an existing field</b>	In the list of fields, select the field that you want to use.
<b>Create a new field</b>	<ol style="list-style-type: none"> <li>a. In the list of fields, click <b>Create new</b>.</li> <li>b. In the <b>Add field</b> dialog box, in the <b>Field name</b> field, enter a name for the field in a data model.</li> <li>c. In the dialog box, in the <b>Display as</b> list, define a display mode for the picklist. You can display a picklist as a drop-down list or a list of radio buttons.</li> <li>d. In the <b>Picklist</b> options, define choices for the picklist. You can create your own local choices or source them from a data page.</li> <li>e. Click <b>Submit</b>.</li> </ol>

5. Click **Save**.

[Designing questionnaires \(on page 405\)](#)

[Fine-tuning your questionnaire \(on page 426\)](#)

## Formatting a checklist in a questionnaire

Format a checklist in a survey to define the choices that users can select to answer a question. With a checklist, users can select multiple answers to a question. For example, in a questionnaire for job applicants, you can ask users to select the languages that they speak.



**Note:** Only standard Pega Platform applications support questions with a checklist. Applications based on Cosmos React do not display questions with checklists.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Survey</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. If you have multiple questions on the question page, click the question that you want to edit.
3. From the list of question types, select **Checklist**.
4. In the rich text editor, enter your question.
5. Define the choices in the checklist:
  - a. In the **Checklist options** section, click **Add value**.
  - b. In the text field, enter a label for the check box.
6. **Optional:** To control when the checklist displays a choice, define conditions:
  - a. Click **More > Advanced view**.
  - b. In the **Visible when** list, select **When**, and then click **Edit**.



- c. Provide a value to compare at run time, a comparator, and a value to compare against the first value.
  - d. **Optional:** To add more conditions, click **Add a row**.
  - e. If you create multiple rows, define whether an answer needs to meet all or any conditions by selecting either **and** or **or** between the rows.
  - f. Click **Submit**.
7. **Optional:** To store the selected choice under a name that is different from the label, change the reporting value:
- a. Click **More > Advanced view**.
  - b. In the **Reporting value** field, replace the default value with your custom value.  
For example, in a list of languages that users speak, you can use an abbreviation for each language.
8. Click **Save**.

---

[Using advanced questionnaire features in Dev Studio \(on page 436\)](#)

[Defining conditions in the condition builder \(on page \)](#)

[Fine-tuning your questionnaire \(on page 426\)](#)

## Formatting a slider in a questionnaire

Format a slider in a survey to control the range of values from which users select an answer. For example, you can ask how many days users work each week by formatting a slider with the values one through seven.



**Note:** Only standard Pega Platform applications support questions with a slider. Applications based on Cosmos React do not display questions with sliders.



1. Navigate to the questionnaire:

Choices	Actions
<p><b>Open the questionnaire in App Studio</b></p>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<p><b>Open the questionnaire in Dev Studio</b></p>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Survey</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. If you have multiple questions on the question page, click the question that you want to edit.
3. From the list of question types, select **Slider**.
4. In the rich text editor, enter your question.
5. In the **Question settings** pane, format the slider:
  - a. In the **Starting range** field, enter the minimum numerical value for the scale, for example, enter 1.
  - b. In the **Ending range** field, enter the maximum numerical value for the scale, for example, enter 7.
  - c. In the **Interval** field, enter the numerical interval between the possible values on the scale, for example, enter 1.
  - d. **Optional:** To preset the answer to the question, enter a value in the **Default value** field, for example, enter 5.
  - e. **Optional:** To show possible values on the slider, select the **Show captions** check box.
6. Click **Save**.

---

### Related information

[Best practices for creating questionnaires \(on page 423\)](#)

[Designing questionnaires \(on page 405\)](#)

[Fine-tuning your questionnaire \(on page 426\)](#)

## Formatting a radio button matrix in a questionnaire

Format a radio button matrix in a survey to ask a series of related questions that users answer by selecting one option from a repeating group of radio buttons. In a radio button matrix, in each row, users can select



an answer from only one column. For example, you can ask users to rate their pain threshold on a scale of one to five for different activities, such as swimming, running, or walking.



**Note:** Only standard Pega Platform applications support questions with a radio button matrix. Applications based on Cosmos React do not display questions with a radio button matrix.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>, and then click the case type that you want to open.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Survey</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. If you have multiple questions in the question page, click the question that you want to edit.
3. In the list of question types, select **Radio button matrix**.
4. In the rich text editor, enter your question.
5. In the **Columns** section, create columns for the matrix:
  - a. Click **Add column**.
  - b. In the text field, enter a label for the column.
  - c. **Optional:** To add more columns, repeat steps [5.a \(on page 418\)](#) through [5.b \(on page 418\)](#).
6. In the **Rows** section, create rows for the matrix:
  - a. Click **Add row**.
  - b. In the text field, enter a label for the row.
7. **Optional:** To store the selected value from a column under a name that is different from the label, change the reporting value:

- a. Click **More > Advanced view**.
  - b. In the **Reporting value** field, replace the default value with your custom value.
8. **Optional:** To display a row only in selected scenarios, configure visibility conditions:
- a. Click **More > Advanced view**.
  - b. From the **Visible when** list, select **When**, and then click **Edit**.
  - c. Provide a value to compare at run time, a comparator, and a value to compare against the first value.
  - d. **Optional:** To add more conditions, click **Add a row**.
  - e. If you create multiple rows, define whether an answer needs to meet all or any conditions by selecting either **and** or **or** between the rows.
  - f. Click **Submit**.
9. Click **Save**.


---

[Designing questionnaires \(on page 405\)](#)

[Fine-tuning your questionnaire \(on page 426\)](#)

## Formatting a currency question in a questionnaire

Enable users of your application to provide answers that precisely fit their financial business scenarios by creating a question of a currency type. As a result, users can clearly link the number that they provide to the amount of money in a specified currency. For example, if a customer has an international account, by providing a currency in a question about a daily transactions limit, you help users understand how their answers affect their bank account.

 **Note:** Only applications that you build on Cosmos React support currency questions.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <b>Questionnaire</b> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Currency**.
4. In the **Question settings** pane, provide the currency that you want to use:

Actions	Choices
<b>Enter the currency manually</b>	<ol style="list-style-type: none"> <li>a. In the <b>ISO code selection</b> list, select <b>Constant</b>.</li> <li>b. In the <b>Currency ISO code</b>, enter the currency code.</li> </ol>
<b>Source the currency from a property</b>	<ol style="list-style-type: none"> <li>a. In the <b>ISO code selection</b> list, select <b>Property Reference</b>.</li> <li>b. In the <b>Currency ISO code</b> list, select a property that stores the currency that you want to use.</li> </ol>

5. To enable users to provide more precise answers, select the **Allow decimals** check box.

At run time, users can provide answers that include decimals.

6. Click **Save**.

[Making a question required in a questionnaire \(on page 426\)](#)

[Conditionally displaying a question in a questionnaire \(on page 427\)](#)

[Conditionally displaying a question page in a questionnaire \(on page 428\)](#)



## Formatting a decimal question in a questionnaire

Help users of your applications provide detailed and accurate answers that require numbers by creating decimal questions. At run time, users can enter numbers that include decimals after a separator to provide very detailed information. As a result, your application can precisely meet even the most specific business requirements. For example, in a questionnaire about the reimbursement of travel costs, users can provide the precise amounts of their travel expenses.

**Note:** Only applications that you build on Cosmos React support decimal questions.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Decimal**.
4. In the **Question settings** pane, in the **Decimal precision** field, enter an integer that reflects how many places after the decimal separator you want users to be able to enter figures.

At run time, users can provide numbers such as 25.55.

5. For greater clarity, to display a separator in the number, select the **Show thousands separator** check box.
6. Click **Save**.

[Formatting a currency question in a questionnaire \(on page 419\)](#)

[Making a question required in a questionnaire \(on page 426\)](#)

[Conditionally displaying a question in a questionnaire \(on page 427\)](#)



Conditionally displaying a question page in a questionnaire (on page 428)

Validating answers in a questionnaire (on page 431)

## Scoring answers in a questionnaire

Calculate a cumulative score based on your questionnaire results, to generate metrics and trends for a specific focus group. For example, in a questionnaire about satisfaction with customer service, you can assign high scores to better ratings and low scores to poorer ratings to quickly calculate the general satisfaction level of customers.



**Note:** Only standard Pega Platform applications support scoring answers. You cannot configure scoring values in applications based on Cosmos React.

You can configure scoring only for questions that you add to a question page in a questionnaire. Individual questions that you ask by using the Question shape in a case life cycle do not support scores.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. Click a question with one of the following answer formats:
  - **Picklist**
  - **Check box**
  - **Radio button matrix**
3. In the **Question settings** pane, in the **Scoring** section, click the **Properties** icon.
4. In the **Binary** window, assign numeric values to the choices.
5. Click **Submit**.
6. Click **Save**.



When users provide answers, your application calculates the cumulative score for the questionnaire, and saves the score in the `pxCumulativeQuestionScore` property.

[Formatting a picklist in a questionnaire \(on page 413\)](#)

[Formatting a radio button matrix in a questionnaire \(on page 417\)](#)

[Formatting a checklist in a questionnaire \(on page 415\)](#)

## Best practices for creating questionnaires

You can use questionnaires to gather information from users. Apply best practices that can help you create questionnaires that are easy to use and maintain. For example, if you create a questionnaire about customer service satisfaction with short, precise questions grouped into pages based on the service area, you can then quickly adjust the visibility of particular pages based on the user input, and conveniently reuse parts of your questionnaire in the future.

To ensure that your questionnaires achieve the best results, apply the following best practices:

- Ensure that users stay focused and can provide answers efficiently by grouping questions in a logical way.

For example, in a questionnaire that collects information about customer satisfaction with a hotel, ask several questions about the service quality, and then provide a series of questions about the food.

- Limit the total number of questions. You can also divide a questionnaire into smaller questionnaires that your application calls in succession.

Users tend to provide less reliable and specific answers in longer questionnaires.

- Limit the number of questions on each page.

Many questions might cause excessive scrolling.

- Use concise questions.

Users are more likely to provide accurate answers to short, precise questions.

- Ensure that the format of the question matches the type of information that you want to gather.

For example, if you want users to select only one option that describes their favorite type of service that a company offers, create a picklist instead of multiple check boxes. In another example, if you want users to rate their level of satisfaction with a service, create a slider instead of an answer with multiple check boxes.

- Proofread your questionnaire.



Grammatical errors, duplicate questions, and acronyms or terms that you do not explain might discourage users from providing useful answers.

- Ask potential users for feedback on the clarity, length, and ease of use of your questionnaire.
- Ensure that your questionnaires are flexible and relevant in multiple scenarios by applying visibility conditions.

For more information, see [Conditionally displaying a question in a questionnaire \(on page 427\)](#).

- Ensure that users provide answers in the correct format by using validations.

For example, you can process cases faster by ensuring that users can enter only past dates of stay at a hotel. For more information, see [Validating answers in a questionnaire \(on page 431\)](#).

[Designing questionnaires \(on page 405\)](#)

[Creating a questionnaire \(on page 406\)](#)

[Running a questionnaire in a case \(on page 138\)](#)

## Running a questionnaire in a case

Collect a wide variety of information from your application users by running a questionnaire in a case. By providing questions and collecting answers in a structured format, you can quickly incorporate user feedback into a case. Reusing questionnaires speeds up the development of your application and helps you deliver accurate applications that precisely meet the business needs of the users.

Your application runs the questionnaire as a child case in the context of the parent case that includes the Questionnaire step. To save more time, you can propagate data from the parent case to the questionnaire. For example, when a user provides basic information about their stay at a hotel by completing a booking form in a parent case, you can propagate the obtained data to an exit questionnaire. As a result, the questionnaire already includes their basic information, such as the dates of their stay and type of accommodation.

Data propagation also helps you deliver more accurate applications, as you can use data from a parent questionnaire to build logic in a child questionnaire. For example, if a user provides a monthly income greater than a specified amount in a parent questionnaire, you can create visibility conditions in the child questionnaire to skip further questions about their income.

1. Add the Questionnaire shape to the life cycle of your case.
  - a. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
  - b. On the **Workflow** tab, in the **Case life cycle** section, hover over a process in a stage, and then click **Step > More > Automations > Questionnaire**.
  - c. Click **Select**.





2. In the **Create Questionnaire** field, provide a name for the step, for example, `Rate customer service`.
3. In the **Step** properties pane, in the **Select questionnaire** list, select the name of the questionnaire that you want to run as a child case.
4. **Optional:** To reuse information from the parent case, transfer data from the current case to the questionnaire:

Choices	Actions
<p><b>Map fields from the case type to the questionnaire</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>Step</b> properties pane, select the <b>Transfer info to questionnaire case</b> check box.</li> <li>b. In the <b>Transfer information</b> dialog box, in the <b>'From' field</b> column, select the field with the values that you want to transfer.</li> <li>c. In the <b>'To' field</b> column, press the Down arrow key, and then select a destination field to which you want to transfer the value.</li> <li>d. <b>Optional:</b> To add mapped fields to the case type view, in the <b>View</b> section, select the <b>Add mapped fields to [case type view name] view</b> check box.</li> <li>e. Click <b>OK</b>.</li> </ol>
<p><b>Configure advanced mapping by providing a data transform</b></p>	<ol style="list-style-type: none"> <li>a. Switch to Dev Studio.</li> <li>b. In the <b>Step</b> properties pane, select the <b>Transfer info to questionnaire case</b> check box.</li> <li>c. In the list of transfer options, select <b>Existing</b>.</li> <li>d. In the <b>Data transform</b> field, enter the name of the data transform that you want to apply.</li> <li>e. Click <b>Save</b>, and then switch to App Studio.</li> </ol>

At run time, when a user opens the questionnaire that is part of the case processing, the questionnaire already includes the specified data from the parent case.

5. In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the questions in the questionnaire:
  - To route the questionnaire to the user who currently processes the case, select **Current user**.
  - To route the questionnaire to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to complete the questionnaire.
  - To route the questionnaire to a work queue that multiple users share, select **Work queue**, and then, in the **Work queue** field, enter the name of the work queue.
6. Click **Save**.

After you add a Questionnaire step to your parent case type, you can access the data that users provide only after the child case for completing the questionnaire is complete. In the data model of the parent case, the system adds two new properties of a query type: a single property, and a list property. These two new properties source data from the questionnaire child case type. For more information about queries, see [Referencing a data page \(on page 161\)](#).

[Designing questionnaires \(on page 405\)](#)

[Sharing data between parent and child cases \(on page 217\)](#)

[Fine-tuning your questionnaire \(on page 426\)](#)

## Fine-tuning your questionnaire

Refine the way that your questionnaire displays questions and evaluates answers to ensure that you collect valid information in a case. For example, you can define conditions for displaying a question or a question page, validate the answers that the users provide, and define the run-time order of the questions in your questionnaire.

Use the following techniques to fine-tune your questionnaire:

[Designing questionnaires \(on page 405\)](#)

[Changing the run-time order of questions in a questionnaire \(on page 433\)](#)

## Making a question required in a questionnaire

Make a question required to ensure that your questionnaire collects complete information. For example, in a questionnaire about the service at a hotel, if a customer rates service as poor, then a field for additional comments is required.



1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. Click the question that you want to make required.
3. In the **Question settings** pane, in the **Required when conditions** section, click the **Properties** icon.
4. In the **Conditions** dialog box, create a condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.
5. **Optional:** To add more conditions, click **Add a row**.
6. If you create multiple conditions, define whether a question needs to meet all or any conditions by selecting either **and** or **or** between the rows.
7. Click **OK**.

At run time, if the when conditions evaluate to true, the question is required to submit the questionnaire.

[Validating answers in a questionnaire \(on page 431\)](#)

Defining conditions in the condition builder (on page )

## Conditionally displaying a question in a questionnaire

Define conditions to ensure that your questionnaire displays relevant questions, based on the context of the questionnaire participant. For example, in an onboarding application, a question about a preferred office chair is displayed only if the employee does not work remotely.



1. Navigate to the questionnaire:

Choices	Actions
<p><b>Open the questionnaire in App Studio</b></p>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<p><b>Open the questionnaire in Dev Studio</b></p>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. Click the question for which you want to configure visibility conditions.
3. In the **Question settings** pane, in the **Visibility conditions** section, click the **Properties** icon.
4. In the **Conditions** dialog box, create a condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.
5. **Optional:** To add more conditions, click **Add a row**.
6. If you create multiple conditions, define whether a question needs to meet all or any conditions by selecting either **and** or **or** between the rows.
7. Click **OK**.
8. Click **Save**.

At run time, the questionnaire displays the question only if the visibility conditions evaluate to true.

[Conditionally displaying a question page in a questionnaire \(on page 428\)](#)

Defining conditions in the condition builder (on page )

## Conditionally displaying a question page in a questionnaire

Define conditions to ensure that your questionnaire displays related questions that are relevant to a given scenario, based on the context of the questionnaire participant. For example, in an onboarding application, the questionnaire displays questions about preferred office accommodations only if the employee does not work remotely.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. Click the question page for which you want to configure visibility conditions.
3. In the **Page settings** pane, in the **Visibility conditions** section, click the **Properties** icon.
4. In the **Conditions** dialog box, define a condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.
5. **Optional:** To add more conditions, click **Add a row**.
6. If you create multiple conditions, define whether a case needs to meet all or any conditions by selecting either **and** or **or** between the rows.
7. Click **OK**.
8. Click **Save**.

At run time, the question page is displayed only if the visibility conditions evaluate to true.

[Conditionally displaying a question in a questionnaire \(on page 427\)](#)

[Defining conditions in the condition builder \(on page \)](#)

## Displaying questions in a read-only mode

Help users provide accurate answers and speed up the time required to complete a questionnaire by displaying questions in a read-only mode. The read-only mode of questions automates the completion of questionnaires when you reference data for an answer from another questionnaire or question, or when the answer from one question directly relates to other questions. For example, you can create a questionnaire that captures customer's preferences about their bank account. If the customer already provided their monthly income in another case or questionnaire, the application can display a question about their monthly income in the bank account questionnaire in a read-only mode.





**Note:** Only applications that you build on Cosmos React support read-only mode for questions. Questions of the info text and file types do not support read-only mode.

To display a question in a read-only mode, you define a condition in a condition builder. At run time, if the condition evaluates to true, your application displays the question in a read-only mode and prepopulates the answer with the value that the user provided in another case in the field that you use in the condition builder.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the **Question settings** pane, in the **Read-only when conditions** section, click the **Configure** icon.
4. In the **Conditions** dialog box, create a condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

At run time, if the condition evaluates to true, a questionnaire displays the value that a user provided as an answer to the question `What is your monthly income?` in another questionnaire.

5. **Optional:** To add more conditions, click **Add a row**.
6. If you create multiple conditions, define whether a question needs to meet all or any conditions by selecting either **and** or **or** between the rows.
7. Click **OK**.
8. Click **Save**.



Defining conditions in the condition builder (*on page 427*)

Conditionally displaying a question in a questionnaire (*on page 427*)

Conditionally displaying a question page in a questionnaire (*on page 428*)

## Validating answers in a questionnaire

Display an error message in your questionnaire when an answer is invalid, to ensure that the information that users provide is accurate and in an expected format. For example, if a user types letters in a phone number, your application prompts the user to enter only numbers in that field.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. Click the question for which you want to configure a validation.
3. In the **Question settings** pane, in the **Validations** section, click the **Properties** icon.
4. In the **Validate fields based on these conditions** dialog box, in the **Message** field, enter an error message that users receive if the validation fails.
5. Create a condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.
6. **Optional:** To add more rows to the condition, click **Add a row**.
7. If you create multiple rows, define whether an answer needs to meet all or any conditions by selecting either **and** or **or** between the rows.
8. **Optional:** To add more validations, click **Add condition**, and then repeat steps 4 (*on page 431*) through 7 (*on page 431*).
9. Click **OK**.
10. Click **Save**.

If the validation fails at run time, users receive an error message. Users need to correct the answers with failed validation before submitting the questionnaire.

[Making a question required in a questionnaire \(on page 426\)](#)

[Displaying a warning in a questionnaire \(on page 432\)](#)

[Populating a question template \(on page 444\)](#)

[Defining conditions in the condition builder \(on page \)](#)

## Displaying a warning in a questionnaire

Use a warning message to encourage questionnaire participants to provide complete answers. For example, if a user enters their birth date in the future, your application displays a reminder that the date must be in the past and be consistent with the customer being an adult.

1. In the header of Dev Studio, click **Configure > Case Management > Questionnaire**.
2. On the **Questions** tab, click a question that you want to edit.
3. Expand the **Warnings** section.
4. Define the conditions that control when your application displays the warning.

Choices	Actions
<p><b>Use a Boolean expression</b></p> <p><b>Note:</b> Use this approach when users complete your questionnaire offline.</p>	<ol style="list-style-type: none"> <li>In the <b>Visibility</b> list, select <b>Condition (expression)</b>.</li> <li>In the <b>Warning expression</b> field, enter a string expression that evaluates property values in the questionnaire.</li> </ol>
<p><b>Use a when condition</b></p> <p><b>Note:</b> Use this approach when users complete your questionnaire online.</p>	<ol style="list-style-type: none"> <li>In the <b>Visibility</b> list, select <b>Condition (when rule)</b>.</li> <li>In the <b>Warning when</b> field, enter the name of a when condition that evaluates property values in the questionnaire.</li> </ol>



Choices	Actions
<p><b>Build a custom condition that can be a combination of when conditions and Boolean expressions</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>Visibility</b> list, select <b>Condition (expression)</b>.</li> <li>b. Click the <b>Open condition builder</b> icon.</li> <li>c. In the <b>Visible when</b> dialog box, create a condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.</li> <li>d. <b>Optional:</b> To use two conditions, define whether a question needs to meet both or any of the conditions by selecting either <b>and</b> or <b>or</b> between the rows.</li> <li>e. Click <b>Save</b>.</li> </ol>

5. In the **Warning message** field, enter the name of a rule that contains a warning message.

For more information, see *Messages (on page 431)*.

6. Click **Save**.

---

[Validating answers in a questionnaire \(on page 431\)](#)

## Changing the run-time order of questions in a questionnaire

Conditionally skip or revisit questions at run time by adding a branch to a questionnaire. By branching questionnaires, you provide flexibility and promote reuse of existing assets, such as question pages. For example, in a shopping application, you can define a condition that skips a set of questions directed towards new customers if a customer reports making purchases in the past. If the customer indicates an interest in providing feedback about their previous purchases, the questionnaire displays questions about their satisfaction with the service. For a new customer, your application loads a set of questions about how the customer found out about your company.

1. Navigate to the questionnaire:

Choices	Actions
<b>Open the questionnaire in App Studio</b>	<ol style="list-style-type: none"> <li>a. In the navigation pane of App Studio, click <b>Case types</b>.</li> <li>b. In the list of case types, open a case type of the <code>Questionnaire</code> type that you want to edit.</li> </ol>
<b>Open the questionnaire in Dev Studio</b>	<ol style="list-style-type: none"> <li>a. In the header of Dev Studio, click <b>Configure &gt; Case Management &gt; Questionnaire</b>.</li> <li>b. Click the questionnaire that you want to edit.</li> </ol>

2. On the **Questionnaire** tab, click **Add > Logic**.
3. In the **Branch** field, click **Click here to edit page title**, and then enter a unique name for the branch, for example `Returning customer`.
4. In the **Branching conditions** pane, in the **When** section, click **Click to add condition**.
5. Create a condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.
6. **Optional:** To add more rows in the condition, click **Add a row**.
7. If you create multiple rows, define whether an answer needs to meet all or any conditions by selecting either **and** or **or** between the rows.
8. Click **Submit**.
9. In the **Go to** list, define what happens when the condition evaluates to true:
  - To move to a specific questionnaire item, select the item in the list.
  - To exit the questionnaire, select **Exit Survey**.
10. **Optional:** To define more conditions, click **Add path**, and then repeat steps 4 (on page 434) through 9 (on page 434).
11. In the **Otherwise go to** list, define what happens when the condition returns a false value:
  - To move to a specific questionnaire item, select the item in the list.
  - To exit the questionnaire, select **Exit Survey**.
12. Click **Save**.

---

[Designing questionnaires \(on page 405\)](#)

[Creating a questionnaire \(on page 406\)](#)



## Displaying a block of text in a question page

Provide supporting information that helps users answer questions more quickly by displaying a block of read-only text on a question page of your questionnaire. For example, you can add a paragraph to explain to users that a complete answer to the next question helps you improve your application.

1. In the navigation pane of App Studio, click **Case types**.
2. In the list of case types, open a case type of the **Questionnaire** type that you want to edit.
3. On the **Questionnaire** tab, on the question page that you want to edit, add a block of text:

Choices	Actions
<p><b>Create a new block of text</b></p>	<ol style="list-style-type: none"> <li>a. Click <b>Add question &gt; New</b>.</li> <li>b. In the list of question types, select <b>Info text</b>.</li> <li>c. In the text editor, enter informative text.</li> </ol>
<p><b>Reuse an existing block of text</b></p>	<ol style="list-style-type: none"> <li>a. Click <b>Add question &gt; From library</b>.</li> <li>b. In the <b>Questionnaire library</b> dialog box, select a block of text that you want to reuse, and then click <b>Add</b> next to the text.</li> <li>c. Close the dialog box by clicking <b>Add</b>.</li> </ol>



**Note:** In applications based on Cosmos React, you can use only plain text info text. Standard Pega Platform applications support inserting dynamic values and formatting the question text.

4. Click **Save**.

[Fine-tuning your questionnaire \(on page 426\)](#)

[Adding a question to a question page \(on page 409\)](#)

### Visualizing questionnaire data by creating insights in Cosmos React

Analyze and manage data from questionnaires in your application by using insights. For example, you can view the service quality questionnaires from multiple respondents at once, and display their responses as an interactive chart that provides an overview of their satisfaction with the service.



Because a questionnaire is a regular case type, you create and use insights for questionnaires in the same way as for any other case type.

1. In your application, create an insight for your questionnaire case type.  
For more information, see [Creating insights \(on page 406\)](#).
2. **Optional:** To adjust the data display to your individual needs, customize how the insight presents data from your questionnaire.  
For more information, see [Customizing list-based insights \(on page 406\)](#).
3. **Optional:** To visualize data in a user-friendly way that helps you analyze the answers, create a chart that presents answers from your questionnaire.  
For more information, see [Converting tables to charts \(on page 406\)](#).

---

[Creating a questionnaire \(on page 406\)](#)

[Creating insights \(on page 406\)](#)

[Customizing list-based insights \(on page 406\)](#)

[Converting tables to charts \(on page 406\)](#)

## Using advanced questionnaire features in Dev Studio

Use Dev Studio to collect information from users in advanced use cases for which providing questionnaires in a low-code way in App Studio is not sufficient. For example, you can run a single question in a case life cycle to use the question in offline mode, reuse answers in other questions, and import a large number of questions by using an Excel template.

You can complete the following tasks in Dev Studio:

---

[Configuring a data model for a case \(on page 150\)](#)

[Running a questionnaire in a case \(on page 138\)](#)

[Best practices for creating questionnaires \(on page 423\)](#)

## Creating a stand-alone question with a single answer

Support information gathering when a case is offline by creating a stand-alone, single-answer question, which you can reference outside the context of a questionnaire. For example, during a conversation with a customer service representative (CSR), you can ask the customer a single question about their satisfaction with the CSR's service.

1. In the header of Dev Studio, click **Configure > Case Management > Questionnaire**.
2. Click the **Questions** tab.
3. Click **Create question > Browse all types > Single answer > [answer type]**.



4. In the **Name** field, enter a unique name that describes the purpose of the question, for example, enter `Employee address`.
5. **Optional:** To change the scope of the question, expand the **Context** section and provide a different class name, ruleset name, or ruleset version.  
The scope controls which rules can reference your question and which rules your question can reference.
6. **Optional:** To manage your questions efficiently, in the **Category** field, you can enter a category for the question, for example, enter `CustomerService`.
7. In the **Question layout** list, select how your application sources the text of the question:


Choices	Actions
<p><b>Plain text that you provide</b></p>	<ol style="list-style-type: none"> <li>a. Select <b>Plain Text</b>.</li> <li>b. In the text editor, enter the text of the question.</li> </ol>
<p><b>Rich text that you provide</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<ol style="list-style-type: none"> <li>a. Select <b>Rich text</b>.</li> <li>b. In the <b>Question text</b> field, enter your question.</li> <li>c. <b>Optional:</b> To make the output of the question more accurate, apply formatting to your text.</li> </ol>
<p><b>Custom HTML rule</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<ol style="list-style-type: none"> <li>a. Select <b>Custom HTML</b>.</li> <li>b. In the <b>Custom HTML</b> field, enter the name of the rule that contains a custom HTML question.</li> </ol>
<p><b>Note:</b> Use a custom HTML rule only in specific use cases that you cannot complete by reusing the standard rules.</p>	

Choices	Actions
<p><b>A section rule in your application</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<p>a. Select <b>Section</b>.</p> <p>b. In the <b>Section reference</b> field, enter the name of the rule that contains a section that you want to reuse.</p>
<p><b>A paragraph rule in your application</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<p>a. Select <b>Paragraph</b>.</p> <p>b. In the <b>Section reference</b> field, enter the name of the rule that contains a section that you want to reuse.</p>

8. If you provide a text box for the answer, in the **Answer presentation settings** section, define how your questionnaire displays the text box:
  - a. In the **Default** field, enter the default text of the answer.
  - b. In the **Size of text box** field, enter the maximum length for the answer, measured in characters and spaces.
9. If you provide a date picker for the answer, in the **Answer presentation settings** section, in the **Default** field, enter the default date.
10. If you provide a date-time picker for the answer, in the **Answer presentation settings** section, in the **Default** field, enter the default date and time.
11. If you provide a text area for a long answer or a comment, in the **Answer presentation settings** section, define how your questionnaire displays the long answer:
  - a. In the **Default** field, enter the default text of the answer.
  - b. In the **Rows** field, enter the height of the text area, measured in characters.
  - c. In the **Columns** field, enter the width of the text area.
12. If you provide a custom drop-down list for the answer, in the **Drop down type** list, select **Prompt list**, and then provide choices:
  - a. **Optional:** In the **When** column, enter a when condition that controls whether your application displays the choice.
  - b. In the **Display value** column, enter a label for the choice.

- c. In the **Score** column, enter a score value that your application stores when the user selects the response.
  - d. **Optional:** To select an answer by default, in the **selected** column, select the radio button.
  - e. **Optional:** To add another answer, click **Add a row**.
13. If you provide a predefined drop-down list for the answer, in the **Drop down type** list, select **Data page**, and then provide choices:
  - a. In the **Data page** field, enter a data page that returns a list of options.
  - b. **Optional:** To pass parameters that the data page supports, in the **Value** field, enter the parameters.
  - c. In the **Property for value** field, enter a property on the data page that stores the options for the list.
  - d. **Optional:** To customize the labels of the choices, in the **Property for display text** field, enter a property on the data page that contains the labels.
14. If you provide check boxes for the answer, in the **Answer presentation settings** section, in the **Score** field, enter a score value that your application stores when the user selects the check box.
15. If you provide custom radio buttons for the answer, in the **Radio button source** list, select **Prompt list**, and then provide choices:
  - a. In the **When** column, enter a when condition that controls whether your application displays the choice.
  - b. In the **Display Value** column, enter a label for the choice.
  - c. In the **Score** column, enter a score value that your application stores when the user selects the response.
  - d. **Optional:** To select an answer by default, in the **selected** column, select the radio button.
  - e. **Optional:** To add another answer, click **Add a row**.
16. If you provide predefined radio buttons for the answer, in the **Radio button source** list, select **Data page**, and then provide choices:
  - a. In the **Data page** field, enter a data page that returns a list of options.
  - b. **Optional:** To pass parameters that the data page supports, in the **Value** field, enter the parameters.
  - c. In the **Property for value** field, enter a property on the data page that stores the options for the list.
  - d. **Optional:** To customize the labels of the choices, in the **Property for display text** field, enter a property on the data page that contains the labels.
17. If you provide a slider for the answer, in the **Answer** section, define the parameters of the slider:
  - a. In the **Starting range** field, enter the minimum numerical value for the scale, for example, enter 1.
  - b. In the **Ending range** field, enter the maximum numerical value for the scale, for example, enter 7.

- c. In the **Interval** field, enter the numerical interval between the possible values on the scale, for example, enter 1.
  - d. **Optional:** To preset the answer to the question, enter a value in the **Default value** field, for example, enter 5.
  - e. **Optional:** To show possible values on the slider, select the **Show captions** check box.
18. If you provide a custom section for the answer, in the **Answer presentation settings** section, in the **Section reference** field, enter a section that displays fields for the user to enter a response.
19. In the **Map to property** list, select a property in the data model that stores the answer.

 **Note:** Some types of questions do not support property mapping, for example, info text.

20. **Optional:** To ensure that the information that users provide is accurate and in an expected format by displaying an error message in your questionnaire when the answer is invalid, in the **Answer** section, in the **Answer validation** field, enter the name of a validation rule.
21. **Optional:** To encourage questionnaire participants to provide complete answers, configure a warning. For more information, see [Displaying a warning in a questionnaire \(on page 432\)](#).

[Asking a question in a case \(on page 449\)](#)


[Working with offline-enabled mobile apps \(on page \)](#)

[Creating a stand-alone question with multiple answers \(on page 440\)](#)

[Creating questions in bulk \(on page 444\)](#)

## Creating a stand-alone question with multiple answers

Gather information when a case is offline by creating a stand-alone, multi-answer question in Dev Studio. You can reference stand-alone questions outside the context of a questionnaire, for example, while processing an order, you can ask the customer a single question about their satisfaction with the sales application.

 **Note:** Only standard Pega Platform applications support questions with multiple answers. Applications based on Cosmos React display only single-answer questions.

1. In the header of Dev Studio, click **Configure > Case Management > Questionnaire**.
2. Click the **Questions** tab.
3. Click **Create question > Browse all types > Multi answer > [answer type]**.



4. In the **Name** field, enter a unique name that describes the purpose of the question, for example, enter `Customer service rating`.
5. **Optional:** To change the scope of the question, expand the **Context** section and provide a different class name, ruleset name, or ruleset version.  
The scope controls which rules can reference your question and which rules your question can reference.
6. **Optional:** To manage your questions efficiently, in the **Category** field, you can enter a category for the question, for example, enter `SatisfactionRatings`.
7. In the **Question layout** list, select how your application sources the text of the question:

Choices	Actions
<p><b>Plain text that you provide</b></p>	<ol style="list-style-type: none"> <li>a. Select <b>Plain Text</b>.</li> <li>b. In the text editor, enter the text of the question.</li> </ol>
<p><b>Rich text that you provide</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<ol style="list-style-type: none"> <li>a. Select <b>Rich text</b>.</li> <li>b. In the <b>Question text</b> field, enter your question.</li> <li>c. <b>Optional:</b> To make the output of the question more accurate, apply formatting to your text.</li> </ol>
<p><b>Custom HTML rule</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<ol style="list-style-type: none"> <li>a. Select <b>Custom HTML</b>.</li> <li>b. In the <b>Custom HTML</b> field, enter the name of the rule that contains a custom HTML question.</li> </ol>
<p><b>Note:</b> Use a custom HTML rule only in specific use cases that you cannot complete by reusing the standard rules.</p>	

Choices	Actions
<p><b>A section rule in your application</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<p>a. Select <b>Section</b>.</p> <p>b. In the <b>Section reference</b> field, enter the name of the rule that contains a section that you want to reuse.</p>
<p><b>A paragraph rule in your application</b></p> <p><b>Note:</b> Available only in standard Pega Platform applications. Applications based on Cosmos React do not support this option.</p>	<p>a. Select <b>Paragraph</b>.</p> <p>b. In the <b>Section reference</b> field, enter the name of the rule that contains a section that you want to reuse.</p>

8. **Optional:** To format the answers, in the **Answer** section, define how your questionnaire displays the available choices in the question.
9. Define how your application stores the answer in the data model:
  - To not map the answer, in the **Property type** list, select **No Property Mapping**.
  - To store the answer as a single-value property, in the **Property type** list, select **Single Value**.
  - To store the answer as a page list, in the **Property type** list, select **Page List**, and then, in the **Page list** field, enter a page list property to store the answer.

**Note:** The **Property type** list is not available for questions of the table type, because you configure property mapping while configuring each column by completing the **Column property** field.

10. **Optional:** To ensure that the information that users provide is accurate and in an expected format by displaying an error message in your questionnaire when the answer is invalid, in the **Answer** section, in the **Answer validation** field, enter the name of a validation rule.

**Note:** Answer validation is not available for questions of the table type.

- Optional:** To encourage questionnaire participants to provide complete answers, configure a warning. For more information, see [Displaying a warning in a questionnaire \(on page 432\)](#).

### Related information

[Creating a stand-alone question with a single answer \(on page 436\)](#)

[Asking a question in a case \(on page 449\)](#)

[Working with offline-enabled mobile apps \(on page \)](#)

## Reusing answers in questions

Create more personalized and comprehensive questionnaires by reusing the answers to questions in subsequent questions. For example, in a sales application, a question prompts the user to enter their name, and then all following questions can start with the answer that the user provides. You can reuse only answers to questions that already exist in your questionnaire or in the base class in which you create your questionnaire.

**Note:** Only standard Pega Platform applications support reusing answers in questions. Applications based on Cosmos React do not support this option.

- In the header of Dev Studio, click **Configure > Case Management > Questionnaire**.
- Click the questionnaire that you want to edit.
- Click **Add question > New**.
- In the list of question types, select the type of question that you want to create.  
For more information about creating questions, see [Adding a question to a question page \(on page 409\)](#).
- In the text field, enter a question, and then place the cursor where you want to reuse the answer.
- On the text editor toolbar, click the **Insert Property** icon.
- In the **Select from question response** dialog box, select a question from which you want to reuse the answer.  
The question that you select must appear in the questionnaire before the question that you are creating.
- Click **Save**.

At run time, your application populates a property with a value that users provided earlier in the questionnaire.

[Adding a question to a question page \(on page 409\)](#)

## Creating questions in bulk

Develop your questionnaires more quickly by creating multiple questions at the same time. For example, a marketing specialist creates a large number of questions and their variations to collect information for a new campaign, and an application developer uses the questions to develop several questionnaires.

You create questions in bulk by editing a template Excel file, and then uploading the file to your application.

[Adding a question to a question page \(on page 409\)](#)

[Troubleshooting question template errors \(on page 447\)](#)

## Populating a question template

Define single-answer questions that you want to create in bulk by populating a question template. You download the template in your application, and add questions by following the structure of the sample questions that the template contains.

For example, define several questions about the work experience of a job candidate that you can then use to create a questionnaire in your application.

1. Download the question template:
  - a. In the header of Dev Studio, click **Configure > Case Management > Questionnaire**.
  - b. On the **Questions** tab, click **Import questions**.
  - c. Click **Download template**, and then save the **QuestionsTemplate.xlsx** file to your local machine.
2. Open the **QuestionsTemplate.xlsx** file.

The **Sample Questions** tab of the Excel file contains examples of correctly configured questions that you can import.
3. In the **ID** column, enter text that uniquely identifies the question.

If you enter the ID of an existing question, when you import the question template, your application prompts you to update or skip the question.
4. In the **Name** column, enter a label for your question.
5. Provide text for your question:

Choices	Actions
Use plain text	<ol style="list-style-type: none"> <li>In the <b>Question Layout</b> column, select <b>plain</b>.</li> <li>In the <b>Question Plain Text</b> column, enter the question text.</li> </ol>
Use styled text	<ol style="list-style-type: none"> <li>In the <b>Question Layout</b> column, select <b>rte</b>.</li> <li>In the <b>Rich text question content</b> column, enter question text that includes HTML tags.</li> </ol>

6. Configure the answer format for your question:

Choices	Actions
A question that a user answers by attaching a file	<ol style="list-style-type: none"> <li>In the <b>Answer Mode</b> column, select <b>File</b>.</li> </ol>
A question that a user answers in a text area, text box, date, or date and time format	<ol style="list-style-type: none"> <li>In the <b>Answer Mode</b> column, select <b>freeform</b>.</li> <li>In the <b>Answer Freeform Format</b> column, select an option that corresponds to your answer format.</li> </ol>
A question that a user answers by selecting one or more available options	<ol style="list-style-type: none"> <li>In the <b>Answer Mode</b> column, select <b>checkbox</b>.</li> </ol>
A question that a user answers by selecting one of available options	<ol style="list-style-type: none"> <li>In the <b>Answer Mode</b> column, select <b>radio</b>.</li> <li>In the <b>Answer option labels</b> column, enter a comma-separated list of the choices that users can select.</li> <li>In the <b>Answer option values</b> column, enter a comma-separated list of the values that map to the choices that users can select.</li> </ol>

Choices	Actions
	d. In the <b>Radio button label orientation</b> column, choose whether the your application displays choices stacked or side-by-side.
<b>A question that a user answers by selecting one of available options in a drop-down list</b>	a. In the <b>Answer Mode</b> column, select <b>dropdown</b> . b. In the <b>Answer option labels</b> column, enter a comma-separated list of the choices that users can select. c. In the <b>Answer option values</b> column, enter a comma-separated list of the values that map to the choices that users can select.
<b>A question that a user answers in a custom section that you provide</b>	a. In the <b>Answer Mode</b> column, select <b>section</b> . b. In the <b>Section Reference</b> column, enter the name of a section that is in the same class path as the question.

7. In the **Answer layout direction** column, choose whether your application displays the question and answer stacked or side-by-side.
8. To change where your application stores the answer to a question on the clipboard, in the **Answer Property** column, enter the name of a property that is in the same class path as the question.
9. To validate the answer that users provide at run time, in the **Validation** column, enter the name of a validation rule that is in the same class path as the question.
10. **Optional:** To add more questions to the template, repeat steps [3 \(on page 444\)](#) through [9 \(on page 446\)](#).
11. Save and close the file but do not change the file format.

[Troubleshooting question template errors \(on page 447\)](#)

## Importing a question template

Automate the creation of single-answer questions by importing a question template. For example, the author of a marketing campaign provides a list of questions that you can add to a questionnaire in your application to collect information from the customers.



1. In the header of Dev Studio, click **Configure > Case Management > Questionnaire**.
2. On the **Questions** tab, click **Import questions**.
3. Click **Browse**.
4. Navigate to the location of your question template, and then upload the file by clicking **Open**.
5. **Optional:** To make the questions compatible with questionnaires that are outside your work pool, in the **Apply to class** field, provide the name of the class that contains your questionnaires.
6. Click **Next**.

The application validates the entries in your question template.

7. Click **Import**.
8. **Optional:** To review a summary of questions that your application created, updated, or skipped, click **View details**.
9. Click **Done**.

[Troubleshooting question template errors \(on page 447\)](#)

[Adding a question to a question page \(on page 409\)](#)

## Troubleshooting question template errors

When you use a question template to create questions in bulk, you might encounter errors. To resolve common issues that prevent you from importing multiple questions to your application, follow the respective guidelines for each scenario when the importing process fails.

The following are the most common issues that can occur during bulk question imports:

- The template contains the IDs of questions that are already in your application
- The template has an invalid file structure or an invalid file type
- Questions in the template fail validation

## Condition

The process of importing the question template fails and you receive an error message.

## The template contains the IDs of questions that are already in your application

### Solution: Solving duplicate questions

1. In the **Duplicate questions** section of the **Import questions** page, click **View details**.
2. Review the list of questions to confirm that you intentionally added them to the template.
3. Resolve the conflicts:



- To exclude the duplicate questions from the import, select **Skip**.
- To overwrite the existing questions with the information in the template, select **Update**.

If you provide more than one entry with the same ID in the template, your application uses only the information from the first row that it finds.

4. In the **Validate and import** stage, click **Import**.
5. In the **Summary** stage, click **Done**.

## The template has an invalid file structure or an invalid file type

### Solution: Fixing the question template

1. On the **Import questions** page, click **Download template**, and then save the **QuestionsTemplate.xlsx** file to your local machine.
2. Open the **QuestionsTemplate.xlsx** file.
3. Copy the information from your original template to the relevant columns in the file.
4. Save the file but do not change its format.
5. In the **Upload file** stage, click **Browse**.
6. Navigate to the location of your new question template, and then upload the file by clicking **Open**.
7. Click **Next**.
8. In the **Validate and import** stage, click **Import**.
9. In the **Summary** stage, click **Done**.

## Questions in the template fail validation

### Solution: Fixing question configuration

1. In the **Failed validation** section of the **Import survey questions** page, click **View details**.
2. Determine which columns have invalid data or reference an invalid rule by reviewing the error messages for each question.
3. Click **Back**.
4. Update your template to provide the required information.
5. Create the required rules in the same class path as the questions.
6. Click **Browse**.
7. Navigate to the location of your updated question template, and then upload the file by clicking **Open**.
8. Click **Next**.
9. In the **Validate and import** stage, click **Import**.
10. In the **Summary** stage, click **Done**.



## Related information

[Creating questions in bulk \(on page 444\)](#)

[Populating a question template \(on page 444\)](#)

[Importing a question template \(on page 446\)](#)

## Asking a question in a case

Collect a single response from a user who processes a case by using the Question shape in a case life cycle. By isolating individual questions, you can selectively capture user input without the formal structure of a questionnaire. For example, in a home insurance process, a case worker asks a customer what kind of offers they want to receive in the future.

Only standard Pega Platform applications support running a question from the case life cycle outside of a questionnaire. In applications based on Cosmos React, you can run a questionnaire by using the Questionnaire shape. For more information, see [Running a questionnaire in a case \(on page 138\)](#).

1. Add the Question shape to the life cycle of your case.
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. On the **Workflow** tab, in the **Life cycle** section, hover over a process in a stage, and then click **Step > More > Automations > Question**.
  - c. Click **Select**.
2. In the **Answer question** field, provide a name for the step, for example, `Type of offers`.
3. Choose a question to ask:

Choices	Actions
<p><b>Reuse an existing question</b></p>	<ol style="list-style-type: none"> <li>a. In the <b>Question type</b> list, select <b>Rule</b>.</li> <li>b. In the <b>Question rule</b> field, press the Down arrow key, and then select the name of a question.</li> </ol> <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> You cannot select a complex question, because only questionnaires support complex questions.</p> </div>

Choices	Actions
<p data-bbox="233 604 586 638"><b>Create a virtual question</b></p> <div data-bbox="240 701 841 1029" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p data-bbox="250 848 289 882" style="font-size: 24px; border: 1px solid #0070C0; border-radius: 50%; display: inline-block; text-align: center; width: 20px; height: 20px; line-height: 20px;">i</p> <p data-bbox="321 730 797 1003"><b>Note: Virtual questions are not reusable. Use this approach only when you are collecting a text response and the question is specific to the process that contains the Question shape.</b></p> </div>	<p data-bbox="938 226 1430 352">In the <b>Question Info</b> section, you can verify that the question is relevant to the case.</p> <p data-bbox="906 394 1446 520">c. <b>Optional:</b> To ensure that the user provides an answer to the question, select the <b>Required</b> check box.</p> <hr/> <p data-bbox="906 604 1430 638">a. In the <b>Question type</b> list, select <b>Text</b>.</p> <p data-bbox="906 651 1430 772">b. In the <b>Question text</b> field, enter the text of the question that your application displays at run time.</p> <p data-bbox="906 785 1446 953">c. In the <b>Map answer to property</b> field, press the Down arrow key, and then select a property to store the value of the answer that the user provides.</p>

4. In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the question:
  - To route the question to the user who currently processes the case, select **Current user**.
  - To route the question to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to answer the question.
5. **Optional:** To control whether your application displays the question at run time, in the **Display when** field, press the Down arrow key, and then select a when condition.
6. **Optional:** To validate the answer that the user provides, configure the processing to perform:

- a. In the **Pre-question processing activity** and **Post-question processing activity** fields, press the Down arrow key, and then select the name of an activity.
- b. In the **Pre-question javascript function** and **Post-question javascript function** fields, enter the names of functions that the harness of the case includes.



**Note:** Applications based on Cosmos React do not support pre-question and post-question processing. You can run these activities and JavaScript functions only in standard Pega Platform applications.

7. **Optional:** To provide additional comments about the question, in the **Audit note** field, enter some helpful extra text.
8. Click **Save**.

---

[Case life cycle elements \(on page 43\)](#)

[Asking a group of related questions in a case \(on page 451\)](#)

[Running a questionnaire in a case \(on page 138\)](#)

[Designing questionnaires \(on page 405\)](#)

## Asking a group of related questions in a case

Collect information from a user who processes a case by using the Question page shape. By providing a structured format for questions and answers, you can quickly incorporate user feedback into a case. For example, in a car insurance case, a case worker asks a customer a set of questions about the circumstances of an accident, and then uses the answers to create a summary of the insurance claim.

Only standard Pega Platform applications support running a question page from the case life cycle outside of a questionnaire. In applications based on Cosmos React, you can run a questionnaire by using the Questionnaire shape. For more information, see [Running a questionnaire in a case \(on page 138\)](#).

1. Add the Question page shape to the life cycle of your case.
  - a. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
  - b. On the **Workflow** tab, in the **Life cycle** section, hover over a process in a stage, and then click **Step > More > Automations > Question page**.
  - c. Click **Select**.
2. In the **Answer set of questions** field, provide a name for the step, for example, Rate the product.



3. In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the questions on the page:
  - To route the question page to the user who currently processes the case, select **Current user**.
  - To route the question page to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to complete the question page.
4. **Optional:** To control whether your application displays the question page at run time, in the **Display when** field, press the Down arrow key, and then select a when condition.
5. **Optional:** To provide additional comments about the question page, in the **Audit note** field, enter some helpful extra text.
6. Click **Save**.

---

[Adding a question page to a questionnaire \(on page 407\)](#)

[Running a questionnaire in a case \(on page 138\)](#)

## Creating reports based on questionnaires in Dev Studio

Analyze and manage data from questionnaires in your standard Pega Platform applications by creating and configuring report definitions. For example, in a shopping application, you can view the responses from several questionnaires about customer hobbies at once, and display the data as a chart that helps you visualize the distribution of favorite pastimes in different age groups, to help you prepare a new sales offer.



**Note:** In applications based on Cosmos React, you can visualize questionnaire data more conveniently in a low-code way by creating insights on the Explore Data landing page. For more information, see [Visualizing questionnaire data by creating insights in Cosmos React \(on page 435\)](#).

Because a questionnaire is a regular case type, you create and use report definitions for questionnaires in the same way as for any other case type.

1. In your application, create a report definition for your questionnaire case type.  
For more information, see [Creating advanced reports \(on page 406\)](#).
2. **Optional:** To visualize data from the questionnaire and make analyzing the data easier, add a chart that presents the answers from your questionnaire.  
For more information, see [Adding charts in the Report Definition rule form \(on page 406\)](#).

---

[Creating a questionnaire \(on page 406\)](#)

[Report definitions \(on page 406\)](#)



Adding charts in the Report Definition rule form (*on page* )

Creating insights (*on page* )

## Securing case access

Improve the security of your Microjourney by configuring case access. When you define what case types and attachments a user can view and edit, you ensure that only the appropriate case workers can acquire specific information about your business processes. For example, you can ensure that only managers can view documents that include sensitive customer data, and that only case workers with the right permissions can work on cases that process this data.

### Related information

Managing work across your team (*on page* )

[Managing concurrent case access](#)

### Configuring case type access

The Access Manager **Work & Process** tab provides information about the authorizations to operations in the application that users have, based on case types (work classes) in the application. Use Access Manager to configure the actions that access groups can do with case types.

You can modify authorization for the following user operations on instances of the expanded case type (work class):

- **Open** – View and open cases.
- **Run Reports** – Run reports that reference cases.
- **Modify** – Edit and save cases.
- **Delete** – Delete cases.
- **View History** – View case history class types.
- **Perform** – Access and work on cases of this type that are assigned to other operators.
- **Process Flows** – Run the flow, if defined for the case type.
- **Flow Actions** – View and click items in the **Action** menu.

---

Access Manager (*on page* )

[Managing concurrent case access](#)

## Viewing authorizations for case type items in a single access group

From the Access Manager you can view settings for access groups other than your own. This allows you to review the authorization settings for access groups and case type items.



You can see the following information:

- Operators in this access group who have full access to this action, flow, or flow action in the case type.
- Operators in this access group who have no access to this action, flow, or flow action in the case type.
- A combination of conditional access and possibly no access to this action, flow, or flow action among operators in the access group.

To view the authorizations for items in a case type in a single access group, complete the following steps:

1. In the header of Dev Studio, click **Configure > Org & Security > Access Manager > Work & Process**.
2. Click **Application**.
3. Select or clear the applications that you want to see.
4. Click **Apply**.
5. Select an access group from the **Access Group** list.
6. View or edit the case type.
 

The page displays the case types and their access status for the access group.

  - a. Click an individual case type or role to edit it.
  - b. Expand a case type category to see authorization for each user operation in the case type.
7. To generate a PDF file that shows all the authorizations for each user operation in the case type, click **Export authorizations**.

---

Access Manager (on page )

[Configuring case type access \(on page 453\)](#)

## Editing authorizations for case type items in a single access group

In Access Manager you can edit and change the authorization for case type items in a single access group.

**Note:** You cannot edit standard Pega Platform roles.

1. In the header of Dev Studio, click **Configure > Org & Security > Access Manager > Work & Process**.
2. Click **Application**.
3. Select or clear any applications that you want to edit.
4. Click **Apply**.
5. Select an access group from the **Access Group** list.

The page displays the case type and the access status for the access group.



6. Expand a case type to see the authorization for each user operation in the case type.
7. In the column for the access role to be authorized, click the **Access** icon.
8. Select one of the following access types.
  - **Full Access** – Grant operators with this role full access to the item.
  - **No Access** – Deny operators with this role access to the item.
  - **Conditional** – Grant access based on an Access When condition. Select an Access When condition under which an operator in the access group can access the item.
9. Click **OK**.

Access Manager (on page )

[Configuring case type access \(on page 453\)](#)

## Viewing authorizations for case type items in all access groups

From the Access Manager allows you to view the authorization settings for case type items. You cannot modify authorizations when viewing all access groups.

The view shows you only the following information:

- Operators in this access group have full access to this action in the case type.
- Operators in this access group have no access to this action in the case type.
- Operators in this access group who have a combination of conditional access and possibly no access to this case type.

To view the authorizations for items in a case type in all access groups, complete the following steps:

1. In the header of Dev Studio, click **Configure > Org & Security > Access Manager > Work & Process**.
2. Click **Application**.
3. Select or clear the applications that you want to see.
4. Click **Apply**.
5. In the **Access Group** field, select **All Access Groups**.
6. View or edit the case type.
 

The page displays the access groups for the application and their access status for each.

  - a. Click an individual case type or role to edit it.
  - b. Expand a case type to see authorization for each user operation in the access groups.
7. To generate a PDF file that shows all the authorizations for each user operation in the case type, click **Export authorizations**.

Access Manager (on page )

[Configuring case type access \(on page 453\)](#)



## Viewing authorizations for case type flows and flow actions in a single access group

Access Manager displays all flows and flow actions defined for each case type in the selected application or applications. This view allows you to review the authorization settings for flows and flow actions.

You can see the following information:

- Operators in this role have full access to this process flow, or flow action in the case type.
- Operators in this role have no access to this process flow, or flow action in the case type.
- Some combination of conditional and possibly no access to this process flow, or flow action among operators in the access group.

To view the authorizations for flows and flow actions in a case type in a single access group, complete the following steps:


1. In the header of Dev Studio, click **Configure > Org & Security > Access Manager > Work & Process**.
2. Click **Application**.
3. Select or clear any applications that you want to view.
4. Click **Apply**.
5. Select an access group from the **Access Group** list.
6. Expand a case type to view authorizations for each user operation in the case type.
7. Click **Process Flows** or **Flow Actions** to expand the list.
8. To generate a PDF file that shows all the authorizations for each user operation in the case type, click **Export authorizations**.

---

Access Manager (on page )  
[Configuring case type access \(on page 453\)](#)

## Editing authorizations for case type flows and flow actions in a single access group

In Access Manager you can edit and change the authorization for case type flows and flow actions in a single access group.

 **Note:** To edit flows and flow actions, be sure that their containing ruleset is unlocked.

1. In the header of Dev Studio, click **Configure > Org & Security > Access Manager > Work & Process**.
2. Click **Application**.
3. Select or clear any applications that you want to edit.



4. Click **Apply**.
5. Select an access group from the **Access Group** list.
6. Expand a case type to see authorizations for each user operation in the case type.
7. Click **Process Flows** or **Flow Actions** to expand the list.
8. In the column for the access role to be authorized, click the **Access** icon.
9. Select one of the following access types:
  - **Full Access** – Grant operators with this role (as well as all other operators in the selected access group) full access to the item.
  - **No Access** – Deny operators with this role access to the item.
  - **Conditional** – Grant access based on an Access When condition. Select an Access When condition under which an operator in the access group can access the item.
10. Click **OK**.

---

Access Manager (on page )

[Configuring case type access \(on page 453\)](#)

### Controlling access to case attachments

Enforce security requirements and prevent users from inadvertently modifying case data by controlling access to attachments in a case. By applying security policies to case content, you ensure that users only interact with attachments that are appropriate for their role in a business process. Consider a scenario in which a loan applicant provides a set of documents that includes an ID scan, proof of income, and a financial history with previous loans. After you restrict user actions for attachments, only the customer service representatives (CSRs) who work on loan requests can view the content attached to the case, instead of users with other roles in your application. Additionally, you can specify that only a manager can view the customer's financial history. To ensure that users always have access to all the documents required for a case resolution, you can also prevent users from deleting any of the attachments.

You secure case attachments by first categorizing attachments and then defining what actions users can perform on each category. For example, you can allow users to edit documents to ensure that the case includes all relevant and up-to-date information.

---

[Extension points and supporting rules for attachments \(on page 461\)](#)

[Setting required attachments for stage entry \(on page 59\)](#)

[Organizing and managing access to case attachments](#)

### Categorizing case attachments

To facilitate a more efficient resolution of your business goals, ensure that users have relevant data during case processing by selecting attachments categories that a case type supports. When you categorize attachments, you define business classifications to provide access control policies that are specific to each



attachment category. For example, first add a Classified category so that later you can define a group of users who can access the content of this category, such as managers.

You can save time by selecting an existing category or create new categories to address your business requirements. You can also specify whether an attachment of a specific category is required to resolve a case. Consider a scenario for a loan application case type, in which a resolution is possible only when a case contains a document with the proof of income of an applicant.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **Attachment categories**.
3. Click **Add attachment category**.
4. In the **Category name** field, identify an attachment category.
  - To create a new attachment category, enter a unique name.
  - To use an existing attachment category, press the Down arrow key, and then select the name of the attachment category.
5. **Optional:** To ensure that a case contains at least one attachment in the attachment category when a user submits the case, select the **Required for resolution** checkbox.
6. Click **OK**.
7. Click **Save**.

The system applies your changes to all unresolved cases. Users can select the attachment category when they add an attachment to a case.

[Empowering knowledge workers \(on page 348\)](#)

[Improvements in the utilities pane \(on page \)](#)

## Attachment types

Attachment types define the media and format of items that users attach to cases in an application.

The following standard attachment types are supported:

- **File** — Uploaded files of any type

By default, a file attachment is saved as an instance of the `Data-WorkAttach-File` class, which maps to the `pc_data_workattach` database table. Optionally, application users can associate a category with each file attachment. The category describes the purpose or contents of the attachment.

- **Note** — Notes that contain only text



You can type or paste text into a note attachment. A note attachment is saved as an instance of the `Data-WorkAttach-Note` class. You cannot edit a note attachment after it is saved.

- **Screenshot** — Screen capture images

Your application can include a window capture feature, so that users can capture a picture of any desktop window. Users can permanently save this image as an attachment to a case (an instance of the `Data-WorkAttach-ScreenShot` class).

- **URL** — A single URL

A URL attachment links the case to a web page or other object that is accessible with the HTTP protocol. A URL attachment is saved as an instance of the `Data-WorkAttach-URL` class.

- **Content** — Content that is saved in an external Enterprise Content Management (ECM) system

You can configure your application to store attachments of type **file** in an external ECM system by using the CMIS protocol. A content attachment is saved as an instance of the `Data-WorkAttach-ECM` class.

---

[Restricting user actions for case attachments \(on page 459\)](#)

[Attaching content to a case \(on page 101\)](#)

[Limitations for attachments in a file storage repository \(on page 403\)](#)

## Restricting user actions for case attachments

Improve case security by ensuring that users can interact with content that is appropriate for their role only. When you define conditions or privileges in an attachment category, you control which actions a user can take on a case attachment. For example, when working with a loan request case type, only a manager can access documents that contain customer's sensitive data.



**Note:** At run time, restricting actions works for attachments of file and URL types only.

1. In the navigation pane of Dev Studio, click **App**.
2. In the class of the case type that you want to configure, expand the **Process** section.
3. Click **Attachment Category**, and then click the name of the attachment category that you want to configure.
4. Identify the case attachments types that your attachment category supports:
  - a. Click the **Availability** tab.
  - b. Select the check box next to one or more relevant attachment types.
5. On the **Security** tab, restrict user actions on the attachment types:

**Choices**

**Restrict user actions accordingly to a privilege**

**Actions**

- In the **Access control list by privilege** section, in the **Privilege** field, enter a privilege that you want to use to grant user actions.
- Select a check box in one or more columns, based on the user operations that this privilege grants.
- To add more privileges and specify other actions for each privilege, click **Add privilege**, and then repeat steps [5.a \(on page 460\)](#) and [5.b \(on page 460\)](#).

At run time, users with the specified privileges can perform actions that you assign to each privilege.


**Restrict user actions accordingly to a when condition**

- In the **Access control list by When Rule** section, in the **When** field, enter a when condition that you want to use.
- Select a check box in one or more columns, based on the user operations that this privilege grants.
- To add more when conditions, click **Add when**, and then repeat steps [5.a \(on page 460\)](#) and [5.b \(on page 460\)](#).

At run time, if a when condition evaluates to true, users can perform actions that you associate with the when condition.



**Note:** The absence of a privilege or when condition does not automatically restrict a user operation. For example, if you define a condition that allows users to create attachments,

 ensure that you also define another condition that restricts editing, viewing, and deleting attachments.

6. **Optional:** To allow users to choose which teams can access the attachments that the users provide, select the **Enable attachment-level security** check box.
7. Click **Save**.

A user can perform an action only when all the when conditions return a true value, the user belongs to the required team, and the user holds at least one of the required privileges.

### Related information

[Setting required attachments for stage entry \(on page 59\)](#)

[Empowering knowledge workers \(on page 348\)](#)

[Creating a rule specialized by circumstance \(on page \)](#)

## Extension points and supporting rules for attachments

You can use extension points, system settings, and standard rules to customize the processing that occurs when you add an attachment to a case. For example, you can scan an attachment for viruses.

The following rules support attachments in a case:

### **Data-WorkAttach-File.CallVirusCheck**

An activity that you can override to integrate your application with an external virus-checking module, such as a Java routine or Windows DDL file.

### **Work-.EditAttachment**

A flow action that you can add to a process so that users can open and edit file attachments in a case. When you submit the form for this flow action, your application creates a new version of the attachment.

### **Initialization/MaximumFileUploadSizeMB**

An attribute that you can set in the *MaximumFileUploadSizeMB* dynamic system setting that you create to change the default size limit of 1 GB for attachments. For more information on how to create a dynamic system setting, see [Creating a dynamic system setting \(on page \)](#).

The following figure shows a sample customization of the *MaximumFileUploadSizeMB* dynamic system setting:

After

Short description \*

prconfig/initialization/maximumfileuploadsizeemb/default

Owning Ruleset

Pega-Engine

Setting Purpose

prconfig/initialization/maximumf

you create the dynamic system setting, the **Value** field helps you set the maximum attachment size. The following figure shows a dynamic system setting that allows a maximum attachment size of 512 MB:

**Edit Dynamic System Settings: prconfig/initialization/maximumfileuploadsizeemb/default**

ID: Pega-Engine • prconfig/initialization/maximumfileuploadsizeemb/default RS: UnCharted [Edit]

Settings History

Value

512

### pc\_data\_workattach

The default table in the database that you can query to retrieve case attachments. You can also store case attachments in external Content Management Interoperability Services (CMIS) systems or web storage providers.

### GetAttachmentReference

An activity that you can override to customize attachment download.

### pyLinkPulseAttachmentToCase

A when rule that you can override and save to your ruleset to enable or disable linking attachments from a Pulse post to a case.

[Attaching content to a case \(on page 101\)](#)

## Controlling access to case attachments

Enforce security requirements and prevent users from inadvertently modifying case data by controlling access to attachments in a case. By applying security polices to case content, you ensure that users only interact with attachments that are appropriate for their role in a business process. Consider a scenario



in which a loan applicant provides a set of documents that includes an ID scan, proof of income, and a financial history with previous loans. After you restrict user actions for attachments, only the customer service representatives (CSRs) who work on loan requests can view the content attached to the case, instead of users with other roles in your application. Additionally, you can specify that only a manager can view the customer's financial history. To ensure that users always have access to all the documents required for a case resolution, you can also prevent users from deleting any of the attachments.

You secure case attachments by first categorizing attachments and then defining what actions users can perform on each category. For example, you can allow users to edit documents to ensure that the case includes all relevant and up-to-date information.

---

[Extension points and supporting rules for attachments \(on page 461\)](#)

[Setting required attachments for stage entry \(on page 59\)](#)

## Interacting with cases at run time

Attain your business objectives by creating and managing cases in your application. Learn how to use tools that enable collaborative work on cases.

---

### Related information

[Enhance the case processing experience in Theme Cosmos applications](#)

### Creating and managing cases

Provide users of your application with the ability to manage work by creating, working through, and resolving cases in the portal that they log in to. For example, in a banking application, a CSR can create cases to open a new bank account for a customer or investigate and resolve a customer inquiry. After creating a case, case workers can use additional tools, such as an option to follow and tag a case or the ability to process cases in bulk, which makes the processing more effective. You can also enable users to customize the dashboard that they see after they log in to your application.

Pega Platform™ offers a wide range of tools to help users manage and resolve work in a convenient and effective way. The availability of the tools varies depending on the portal that users can access. When you create a new application, Pega Platform automatically generates the first portal for you. Depending on your UI architecture, the default portals are User Portal for Theme Cosmos applications, or Web Portal for Cosmos React applications.

For more information about the types of applications that you can build in Pega Platform, see [Theme UI-Kit](#), [Theme Cosmos \(on page 463\)](#), and [Cosmos React \(on page 463\)](#).

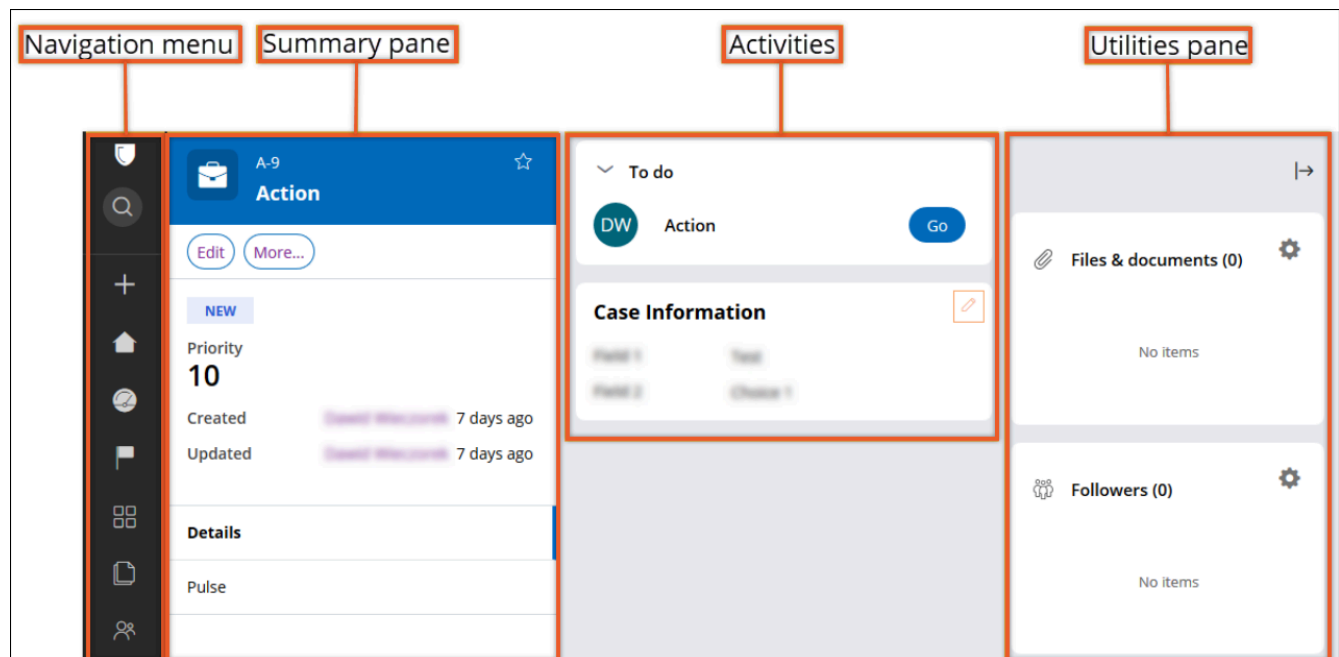


## Theme Cosmos applications

In the recommended Theme Cosmos Pega provides a redesigned, fully-customizable UI in one portal that offers the most comprehensive list of features. For more information, see [Enhance the case processing experience in Theme Cosmos applications](#).

For example, in the Theme Cosmos applications, you can use the preview pane to see the most important case information, collaborate by using Pulse, and interact with case widgets. You can preview cases and documents directly from their links and open related cases in new browser tabs to keep the context of your current work. You can conveniently switch between cases with your browser's back button and follow navigable breadcrumbs to trace relationship between related work. If you need to return to a task later, you can bookmark cases, new spaces, or important documents, and access them again in the Recents menu. You can maximize the work area by hiding Spaces and Documents sections, and expanding them only when necessary. An expandable pane on the right side holds tabs for metadata details and utility widgets, and case information is presented in the following ways:

- The **Navigation** menu on the left allows for easy access to the main pages.
- Case data and related objects display in the **Summary** pane, which has a customizable interface.
- The work area features activity and life-cycle tasks, such as ad-hoc, suggested, or completed tasks. Any collaboration or message history captured in Pulse also appears in the work area.
- The expandable **Utilities** pane displays widgets for participants, attachments, and tags.



UI of an example case created in User Portal

For greater flexibility and better user experience, you can customize User Portal for specific personas and enable case workers to personalize the case UI at run time. For more information, see [Configuring portals](#)



([on page 463](#)), Organizing the contents of a portal ([on page 464](#)), and Summary panel views ([on page 465](#)).

## Cosmos React applications

The Pega Cosmos design system offers the most important information to users at a glance. The main visual areas of a Cosmos case are similar to the Theme Cosmos case UI and include:

- The global **Navigation** menu on the far left, which contains all navigation elements within a single global navigation component, instead of the traditional application header.
- The **Summary** pane, which contains critical information about the case. The size and shape are purposefully similar to a mobile device. The placement of the **Summary** pane ensures that the most critical information of the record is displayed in the upper-left corner of the screen. On any new screen, users who read left-to-right instinctively look to the upper-left part of the screen to understand their context and get their bearings.
- The **Activities** area.
- The **Utilities** pane on the right, which provides space for a number of widgets that help users process a case more efficiently and collaboratively.

For information about portals and the tools that you can use in your Cosmos React application, see [Configuring portals \(on page 463\)](#), [Organizing the contents of a portal \(on page 464\)](#), [Widgets in Cosmos React \(on page 465\)](#), and [Out-of-the-box portal features in the Cosmos React applications \(on page 468\)](#).

---

### Related information

[Application UI setup \(on page 463\)](#)

[Cosmos React \(on page 465\)](#)

[Theme Cosmos \(on page 463\)](#)

[Setting up the Constellation UI Service in Pega Platform \(on page 463\)](#)

[Pages \(on page 463\)](#)

[Configuring a full view \(on page 463\)](#)

## Out-of-the-box portal features for creating and managing cases

Out-of-the-box features in portals help users plan work efficiently and complete common management tasks. Pega Platform™ provides a variety of ready-to-use custom pages that cover common work scenarios. For example, you can set up a **Teams** page to help users quickly find all the teams with which they work, or an **Explore Data** page that features reporting tools for data and case types.



## Out-of-the-box portal features in the Theme Cosmos applications

Feature	Description
<b>Create</b>	<p>Enables users to start a new case, which is an instance of one of the case types that is defined in your application and added to the <b>Create</b> menu.</p> <p>For more information, see <a href="#">#unique_270 (on page )</a>.</p>
<b>Home</b>	<p>Provides users with a quick overview of the application. You can populate the home page with out-of-the-box widgets, such as <b>Pulse</b>, <b>App announcements</b>, and <b>To do</b> list, as well as fields and views that best meet the needs of your business.</p> <div data-bbox="824 961 1461 1335" style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p><b>Note:</b> Unlike a traditional Pega Platform dashboard that the end user can customize, the Home page in a Theme Cosmos application is a curated experience. Consequently, it is the application designers who specify what users see when they log in.</p> </div> <p>For more information, see <a href="#">Home (on page )</a>.</p>
<b>Explore Data</b>	<p>Contains react-based reporting tools for data and case types.</p> <p>For more information, see <a href="#">Explore Data (on page )</a> and <a href="#">Visualizing data with insights (on page )</a>.</p>
<b>Search</b>	<p>Helps you control what data is available for the Search and Reporting Service (SRS).</p>

Feature	Description
	For more information, see Search and Reporting Service (on page ).
<b>Dashboard</b>	<p>Provides quick insight into key work metrics. You can customize the dashboard by changing its layout and adding widgets, such as charts or shortcuts to reports and frequently used resources.</p> <p>Dashboards are customizable by end users. For more information, see <a href="#">Dashboard management (on page 494)</a>.</p>
<b>Spaces</b>	<p>Lists and helps users manage collaboration spaces.</p> <p>For more information, see <a href="#">Collaboration with users by using Spaces (on page 510)</a></p>
<b>Reports</b>	<p>Provides quick access to reports in the application.</p> <p>For more information, see Report Browser (on page ) and Editing a report (on page ).</p>
<b>Documents</b>	<p>Lists application documents that users create or follow.</p> <p>For more information, see <a href="#">Collaborating on shared content by using Documents (on page 527)</a>.</p>
<b>My Work</b>	Provides quick access to cases and assignments of the user.
<b>Pulse</b>	<p>Displays users' message feed from the <b>Pulse</b> gadget.</p> <p>For more information, see <a href="#">Configuring Pulse for case types (on page 220)</a> and <a href="#">Collaborating with users by using Pulse (on page 499)</a>.</p>

Feature	Description
<b>Tags</b>	Lists tags that application users add to cases.  For more information, see <a href="#">Case tagging (on page 482)</a> and <a href="#">Creating suggested tags for cases (on page 488)</a> .
<b>Teams</b>	Displays the users in your team (a workgroup in Dev Studio) and the number of their open assignments.  For more information, see <a href="#">Creating a team (on page 492)</a> and <a href="#">Configuring teams (on page 491)</a> .

For more information, see [Configuring portals \(on page 487\)](#), [Organizing the contents of a portal \(on page 488\)](#), [Organizing the main navigation for a portal \(on page 489\)](#), [Creating custom menus \(on page 490\)](#), [Summary panel views \(on page 491\)](#), and [Creating a landing page for an application \(on page 492\)](#).

## Out-of-the-box portal features in the Cosmos React applications

Features	Description
<b>Create</b>	Starts a new case, which is an instance of one of the case types that are defined in your application.  For more information, see <a href="#">#unique_270 (on page 493)</a> .
<b>Home</b>	Provides users with a quick overview of your application. You can populate the home page with out-of-the-box widgets, such as <b>Pulse, App announcements</b> , and <b>To do</b> list, as well as fields and views that best meet the needs of your business.  The Home page in a Cosmos React application is a curated experience. Consequently, the applica-

Features	Description
	<p>tion designers specify what users see when they log in.</p> <p>For more information, see <a href="#">Home (on page )</a>.</p>
<b>Dashboard</b>	<p>A React-based landing page that uses the dashboard template. The Cosmos React dashboard component is a curated experience. Consequently, it is the application designers who specify what users see when they log in. Personalization at run time is not possible.</p> <p>For more information, see <a href="#">Configuring a dashboard (on page )</a>.</p>
<b>My Work</b>	<p>Provides quick access to cases and assignments of users.</p>
<b>Explore Data</b>	<p>Contains React-based reporting tools for case types.</p> <p>For more information, see <a href="#">Explore Data (on page )</a> and <a href="#">Visualizing data with insights (on page )</a>.</p>
<b>Search</b>	<p>Helps you control what data is available for the Search and Reporting Service (SRS).</p> <p>For more information, see <a href="#">Search landing page overview (on page )</a>.</p>
<b>Tags</b>	<p>Lists tags that application users add to cases.</p> <p>The widget requires additional configuration. For more information, see <a href="#">Configuring the Tags widget (on page )</a> and <a href="#">Case tagging (on page 482)</a>.</p>
<b>Followers</b>	<p>Shows the list of users who track the progress of a certain case.</p>

Features	Description
	For more information, see <a href="#">Following a case (on page 489)</a> .
<b>Attachments</b>	Supports adding files and links to assignments.  For more information, see <a href="#">Categorizing case attachments (on page 457)</a> .
<b>Related cases</b>	Shows the list of cases that are related to a certain case.  The widget requires additional configuration. For more information, see <a href="#">Configuring the Related cases widget (on page )</a> and <a href="#">Improvements in the utilities pane (on page )</a> .
<b>Stakeholders</b>	Shows the list of users who are the first points of contact for a certain case.  The widget requires additional configuration. For more information, see <a href="#">Configuring case participants (on page 240)</a> .
<b>Case operator</b>	Includes the name of the user who submits an assignment. The widget does not require additional configuration and is ready to use by default.
<b>Case history</b>	Shows details about assignments in the form of a table. For example, the information includes names of users who perform actions in an assignment. The widget does not require additional configuration and is ready to use by default.
<b>Pulse</b>	Provides users with a message feed in which to communicate with each other. The widget does not require additional configuration and is ready to use by default.

Features	Description
	For more information, see <a href="#">Collaborating with users by using Pulse (on page 499)</a> and <a href="#">Configuring Pulse for case types (on page 220)</a> .

For more information, see [Configuring portals \(on page 220\)](#), [Organizing the contents of a portal \(on page 220\)](#), [Organizing the main navigation for a portal \(on page 220\)](#), [Pages \(on page 220\)](#), [Configuring a full view \(on page 220\)](#), [Widgets in Cosmos React \(on page 220\)](#), and [Feature limitations in Cosmos React \(on page 220\)](#).

### Related information

[Cosmos React \(on page 220\)](#)

[Theme Cosmos \(on page 220\)](#)

[Understanding UI Capabilities in Pega Platform](#)

[Enhance the case processing experience in Theme Cosmos applications](#)

## Creating cases from optional actions

Start one or more cases from the current case by selecting a relevant action from the **Actions** menu. By enabling users to create cases on demand without navigating away from the main case, you make your process more effective, convenient, and suited to your business goals.

For example, in an application to process company purchases, a user can start a new case to create a new vendor, or update vendor data in the system.

1. Log in to an end-user portal, for example, Web Portal.
2. Create or find a case that you want to process.

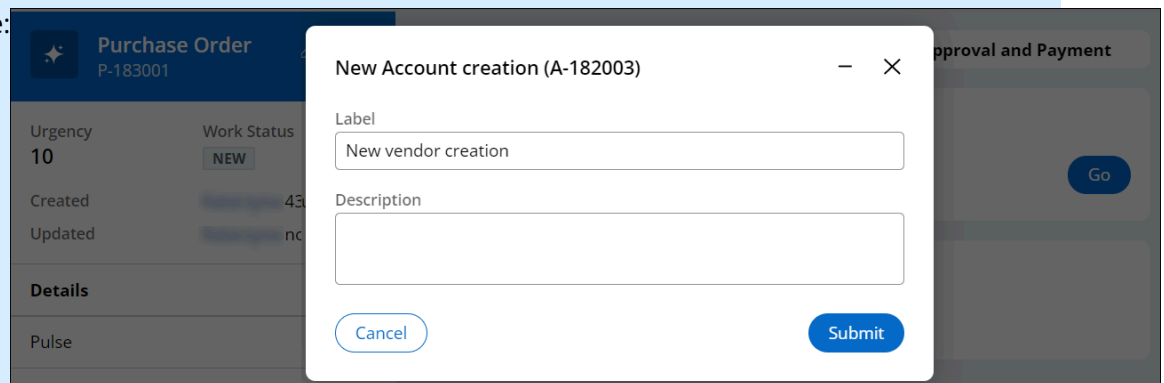


**Tip:** You can locate a case by searching for it or by looking in your worklist or the **Recents** list.

For more information about creating cases, see [#unique\\_270 \(on page 220\)](#).

3. In the **Tools** section, click **Actions** > *optional action name*.

The system runs the selected child case. If the new case has the Create stage with steps configured, you can see a modal dialog box in which you can enter the new case details, as in the following example:



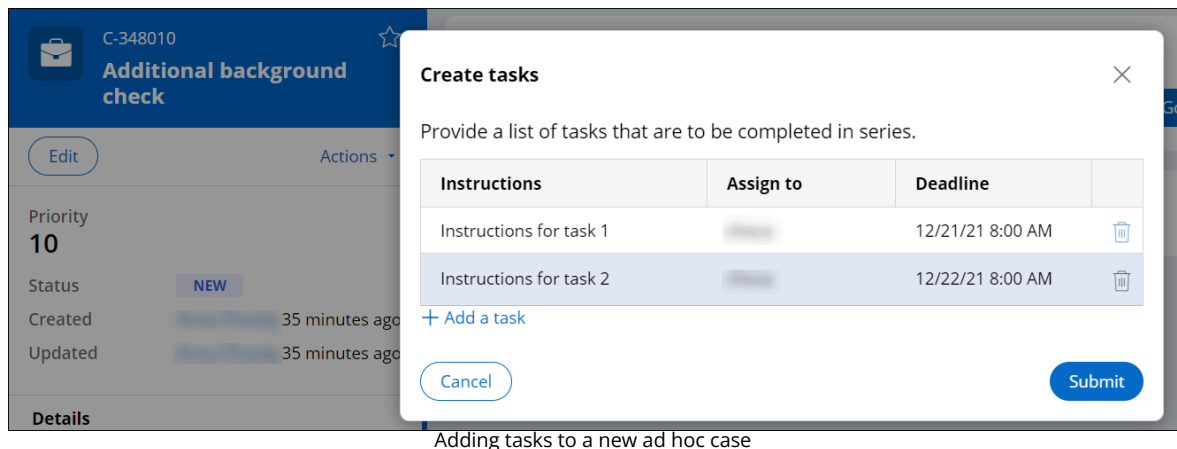
Child case created from the current case

## Creating ad hoc cases

You can handle business exceptions or track related tasks in your current case by creating an ad hoc case. Using an ad hoc case, you can resolve exceptional processes without modifying the existing case structure because ad hoc cases are not instances of a case type. For example, a case worker, such as a human resources employee, can create an ad hoc case when an application from a job candidate requires an additional background check.

1. Log in to an end-user portal, for example, User Portal.
2. In the **My Cases** widget, in the **Create an ad hoc case** text field, enter a name for your ad hoc case.
3. Click **Quick create**.
4. Add tasks to the ad hoc case:
  - a. In the **My cases** list, double-click the name of the newly created ad hoc case.
  - b. In the case Summary pane, click **Actions**, and then select **Create tasks**.
  - c. In the **Ad hoc case dashboard** section, click **Add a task**, and then specify instructions for the task, an assignee, and a deadline.





Adding tasks to a new ad hoc case

d. Click **Submit**.

Your application displays the tasks in the **To do** list in the **Activities** area of the case. Users can add notes upon completing the task. If you create multiple tasks in an ad hoc case, your application displays the next task after the user resolves the current task.

[Empowering knowledge workers \(on page 348\)](#)

[Views for cases \(on page 168\)](#)

## Transferring an assignment

You can transfer an assignment from one user to another. For example, you can transfer an assignment to a team member with more experience or less workload. By reassigning work to other members on your team, you can resolve cases more quickly and effectively.

The Transfer assignment optional action enables users to manually transfer an assignment to another user, a work queue, or team.



**Note:** Ensure that your application can use Theme Cosmos or Cosmos React. For more information, see [Enabling hybrid mode \(on page 348\)](#) and [Setting up the Constellation UI Service in Pega Platform \(on page 168\)](#). For applications that use Theme UI-Kit, see [Transferring an assignment](#).

1. Log in to an end-user portal, for example Web Portal.
2. Find and open a case that you want to open.
3. In the **Tools** section, click **Actions > Transfer assignment**.

4. In the **Activities** pane of the case, in the **Transfer assignment** section, select the new assignee.
5. Click **Transfer**.

## Reopening a resolved case

Reopen a resolved case if users resolved the case prematurely or erroneously. After reopening the case, you can process and resolve the case again.

1. Log in to an end-user portal, for example, Web Portal.
2. Find, and then open a resolved case that you want to reopen.  
You can locate a resolved case by searching for it or by looking in your worklist or the **Recents** list.
3. In the **Tools** section, click **Actions > Reopen**.
4. In the **Reopen** section, in the text field, provide the reason for reopening the case.
5. **Optional:** To move the case to a selected stage after reopening it, in the **change stage** list, select the new stage.

After you reopen a case, the case status changes to `open`. If you do not supply a stage to change to when you reopen the case, it reopens in the first stage after the Create stage. Otherwise, it restarts the stage that you selected.

6. Click **Submit**.

---

[Changing case statuses \(on page 318\)](#)

## Bulk processing

Process cases in bulk to save time and resolve work faster. Bulk processing is less error-prone than individual case processing.

For example, managers can use bulk processing to transfer a large number of cases to a specific user for processing, update the urgency for multiple cases, or send out a message about many cases to a specific user.

Developers can instantly apply common flow actions to many cases. For example, they can adjust service-level agreement times and add a note to all cases at once.



**Note:** Cosmos React applications support bulk actions only for tables and lists that appear on landing pages and case pages. For more information, see [Configuring a list-based landing page \(on page \)](#) and [Configuring tables \(on page \)](#).

## Related information

[Understanding case hierarchy \(on page 14\)](#)

[Validating case data \(on page 164\)](#)


[Queue processor rules \(on page \)](#)

[Pega-ProCom agents \(on page \)](#)

## Creating cases in bulk

Create cases in bulk to define case details, such as labels and work statuses, in one step without manually interacting with individual forms. For example, in an insurance application, you can save time by creating multiple cases for home, car, and life insurance in bulk.

**Note:** Cosmos React applications support bulk actions only for tables and lists that appear on landing pages and case pages. For more information, see [Configuring a list-based landing page \(on page \)](#) and [Configuring tables \(on page \)](#).

1. In an end-user portal, for example, the Case Manager portal, in the rightmost end of the navigation pane, click the user icon (for example ) , and then select **Bulk actions**.
2. In the **Bulk** section, click **Create**.
3. In the **Case Type** list, select the name of a top-level case type.  
Pega Platform supports bulk creation only for parent case types.
4. Click **Add Case**, and then define values for your cases:
  - a. In the **Label** field, enter the name of the case that is visible in worklists and forms.
 

The application stores labels in the *Work-.pyLabel* property.
  - b. In the **Description** field, enter the purpose of the case.
 

The application stores descriptions in the *Work-.pyDescription* property.
  - c. In the **Status** list, select the initial status of the case.
 

The application stores statuses in the *Work-.pyStatusWork* property and updates them automatically when case workers process a case.
5. **Optional:** To create more cases, repeat step 4 ([on page 475](#)).
6. Click **Create Cases**.

The application creates cases based on the values that you provide, and displays the current status of each case in the **Processing Results** column of the case

list.

Case type\*

ID	Label	Description	Status	Processing Results
I-188001	Car insurance PREMIUM	UPlusTelco car insurance premium offer for a private customer	Open	Succeeded
I-188002	Car insurance regular	UPlusTelco car insurance regular offer for a private customer	New	Succeeded
I-188003	Home insurance PREMIUM	UPlusTelco home insurance premium offer for a private customer	New	Succeeded
I-188004	Home insurance regular	UPlusTelco home insurance regular offer for a private customer	New	Succeeded
I-188005	Life insurance PREMIUM	UPlusTelco life insurance premium offer for a private customer	New	Succeeded
I-188006	Life insurance regular	UPlusTelco life insurance regular offer for a private customer	Open	Succeeded

+ Add Case


Cases that are created in bulk

## Updating cases in bulk

Update cases in bulk to set the values of case labels and descriptions in one step, without manually interacting with individual forms. As a result, you save time and reduce the risk of small errors in case details.

Consider an example of an insurance sales application. After customer service representatives (CSR) make hundreds of insurance offers to existing customers in a month, as part of a new marketing campaign, the company updates all offers from the previous week to the premium level. CSRs can update the labels and descriptions of all the eligible cases in bulk, and save time by avoiding manual changes in each case separately.

**Note:** Cosmos React applications support bulk actions only for tables and lists that appear on landing pages and case pages. For more information, see [Configuring a list-based landing page \(on page \)](#) and [Configuring tables \(on page \)](#).

1. In an end-user portal, for example, the Case Manager portal, in the rightmost end of the navigation pane, click the user icon (for example ) , and then select **Bulk actions**.
2. In the **Bulk** section, click **Update**.
3. In the **Case Type** list, display relevant cases by selecting the name of a case type.
4. **Optional:** To refine the list of cases, add filter criteria:

- a. Click **Add Filter**.
- b. In the **Filter Property** field, enter the name of the property that you want to evaluate in each case.
- c. In the **Condition** list, select a comparator.
- d. In the **Filter Value** field, enter a case-sensitive value to compare with the value of the **Filter Property** field.
- e. **Optional:** To add more filter criteria, repeat steps [4.a \(on page 477\)](#) through [4.d \(on page 477\)](#).
- f. Click **Filter Cases**.

The application displays cases that meet all the filter criteria.


5. Update values in the **Name** and **Description** columns.
6. Click **Update**.

The application updates cases based on the values that you provide, and displays the current status of each case in the **Processing Results** column of the case list.

## Running flow actions in bulk

Run the same flow action for multiple cases at once by using the **Bulk actions** form. You can work on many cases without impacting their path toward resolution. For example, you can extend a service-level agreement (SLA) or transfer a set of assignments to a user.

**Note:** Cosmos React applications support bulk actions only for tables and lists that appear on landing pages and case pages. For more information, see [Configuring a list-based landing page \(on page \)](#) and [Configuring tables \(on page \)](#).

1. In an end-user portal, for example, the Case Manager portal, in the rightmost end of the navigation pane, click the user icon (for example ) , and then select **Bulk actions**.
2. In the **Bulk** section, click **Process**.
3. In the **Process Cases** section, modify the filter criteria to refine the list of cases:  
The default filter criteria return a list of cases with at least one open assignment that is assigned to you.

- a. Click **Add Filter**.
- b. In the **Filter Property** field, enter the name of the property that you want to evaluate in each case.
- c. In the **Condition** list, select a comparator.
- d. In the **Filter Value** field, enter a case-sensitive value to compare with the value of the **Filter Property** field.
- e. **Optional:** To add more filters, repeat steps 3.a (on page 478) through d (on page 478).
- f. Narrow the list of results by clicking **Filter Cases**.

Entering filter criteria

4. Select the cases for which you want to run the flow action:

- To select individual cases, select the check boxes next to the cases for which you want to run the flow action.
- To select all cases, select the **Select all displayed results** check box in the top row of the list of results.



**Note:** If your filter criteria return more than one page of results, to select all of the results, you need to select the **Select all displayed results** check box on each page.

Total results: 75 1 2 Next

<input type="checkbox"/>	Label	ID	Status	Owned By	Last Updated
<input type="checkbox"/>		A-103008	New		10/22/20 7:14 AM
<input type="checkbox"/>		A-103004	New		10/22/20 6:47 AM
<input type="checkbox"/>		A-103001	New		10/22/20 6:37 AM
<input type="checkbox"/>		A-103006	New		10/22/20 6:58 AM
<input type="checkbox"/>		A-103002	New		10/22/20 6:40 AM
<input type="checkbox"/>		A-103007	New		10/22/20 6:59 AM
<input type="checkbox"/>		A-103005	New		10/22/20 6:56 AM
<input type="checkbox"/>		A-103003	New		10/22/20 6:46 AM
<input type="checkbox"/>		A-103009	New		10/22/20 7:23 AM

Selecting all results on multiple pages

5. Click **Select Action**, and then select the flow action to run.

The list of available flow actions depends on the flow actions that apply to the cases that meet your filter criteria.

6. In the flow action dialog box, enter values in the relevant fields.

7. **Optional:** To continue working while your application processes the cases, select the **Bulk process in background** check box.

Some flow actions support background processing, which can help you complete your work faster when you run a flow action for many cases. As a result, an agent queues each request and sends you an email when the processing finishes.

8. Run the flow action by clicking **OK**.

The flow action runs for each case, based on the values that you provide. You can view the status of the action in the **Processing Results** column of the case list.



**Note:** After the flow actions complete, your application does not clear the check boxes next to the cases that you selected. As a result, you can quickly run another flow action for the same cases.

## Troubleshooting errors in bulk processing

When you process cases in bulk, you might encounter errors. To resolve common issues that prevent you from processing multiple cases, follow the respective guidelines for each problem scenario.

The following are the most common issues that can occur during bulk question imports:

- The Operator menu of your portal does not contain the **Bulk actions** option.
- The list of items to bulk process is read-only.
- The **Processing Results** column displays a failed status.
- The **Processing Results** column never displays a status.
- The **Instruction** column displays errors.
- You receive no notification when background processing is complete.
- The list of flow actions in the **Select Action** list is incomplete.

## Condition

Users want to create or update cases, or run flow actions on multiple cases. One of several problems prevents users from processing cases.

## Cause: The Operator menu of your portal does not contain the Bulk actions option.

**Note:** Cosmos React applications support bulk actions only for tables and lists that appear on landing pages and case pages. For more information, see [Configuring a list-based landing page \(on page \)](#) and [Configuring tables \(on page \)](#).

## Solution: Granting the appropriate privileges to your operator.

1. Associate a role with your access group that grants you the *Work-PerformBulk* privilege. For more information, see [Granting privileges to an access role \(on page \)](#).
2. Refresh your browser session.

## Cause: The list of items to bulk process is read-only because another user has a lock on some cases.

## Solution: Coordinating your work on cases with other users.

1. Hover over the **Lock** icon in the list of cases to see the name of the user who currently has the case open.
2. Contact the user to coordinate an appropriate time for you to bulk process the case.
3. After the user releases the lock, click **Update** to refresh the list.

## Cause: The Processing Results column displays a failed status.

## Solution: Resolving bulk processing errors.

1. Hover over the failed status to view an error message.
2. Follow the instructions in the error message to resolve the issue.
3. Rerun the bulk action.



## Cause: The Processing Results column never displays a status.

### Solution: Enabling the bulk processing agent.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **SysAdmin** category, and then click **Agents**.
3. In the list of agents on your node, click the *Pega-ProCom* agent.
4. On the **Schedule** tab, in the **Scheduled Agents** list, add the *AgentBulkProcessing* activity, and then select the **Enabled** check box for this activity.
5. In the **Agent-Wide Settings** section, select the **Enable this agent** check box.
6. Click **Save**.

## Cause: The Instruction column displays errors.

### Solution: Resolving assignment errors.

1. In the header of Dev Studio, click **Configure > Case Management > Tools > Work Admin > Worklist Assignment Errors**.
2. In the **Assignments with errors** window, review the list of problem assignments and their corresponding error messages to determine which errors you can clear.

## Cause: You receive no notifications when background processing is complete.

Some flow actions support background processing, which can help you complete your work faster when you run a flow action for many cases. As a result, an agent queues each request and sends you an email when processing finishes. For more information, see Queue processor rules (*on page* [100](#)).

### Solution: Configuring the email account

1. Ensure that the *Default* email account is correctly configured in your application. For more information, see Configuring outbound email in App Studio (*on page* [100](#)).
2. On your Operator form, verify that the **Email** field contains a valid email address.



## Cause: The list of flow actions in the Select Action list is incomplete.



**Note:** The **Select Action** list displays only flow actions that are visible to your current work pool and that have a common ancestor class to the cases that you are processing in bulk.

## Solution: Configuring the display of flow actions.

Complete any of the following steps:

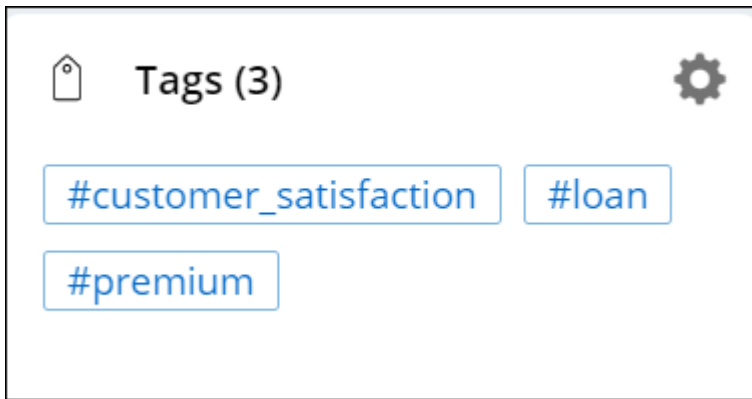
1. Display a flow action that is disqualified from bulk processing:
  - a. In the navigation pane of Dev Studio, click **Records**.
  - b. Expand the **Process** category, and then click **Flow Action**.
  - c. Open the flow action that is missing from the **Select Action** list.
  - d. On the **Action** tab of the Flow Action form, in the **Indicator** section, clear the **Disqualify this action from bulk processing** check box.
  - e. Click **Save**.
2. Display flow actions for cases:
  - a. On the **Bulk actions** page, in the **Filter Property** column, remove the filter criteria that evaluate an assignment.
  - a. Click **Filter Cases**.
  - b. Select the check box next to each case for which you want to run the flow action.
3. Display flow actions and connector actions for assignments:
  - a. On the **Bulk actions** page, in the **Filter Property** column, add filter criteria that evaluate an assignment.
  - a. Click **Filter Cases**.
  - b. Select the check box next to each assignment for which you want to run the flow action.

## Case tagging

You can use tags to classify, manage, and report on tagged cases in a preferred way. Using tags promotes flexibility and collaborative case management.

For example, in a financial application, you can create the tags `premium` and `loan` to flag cases related to specific types of accounts, or tags to flag specific types of issues, for example, `customer_satisfaction`.

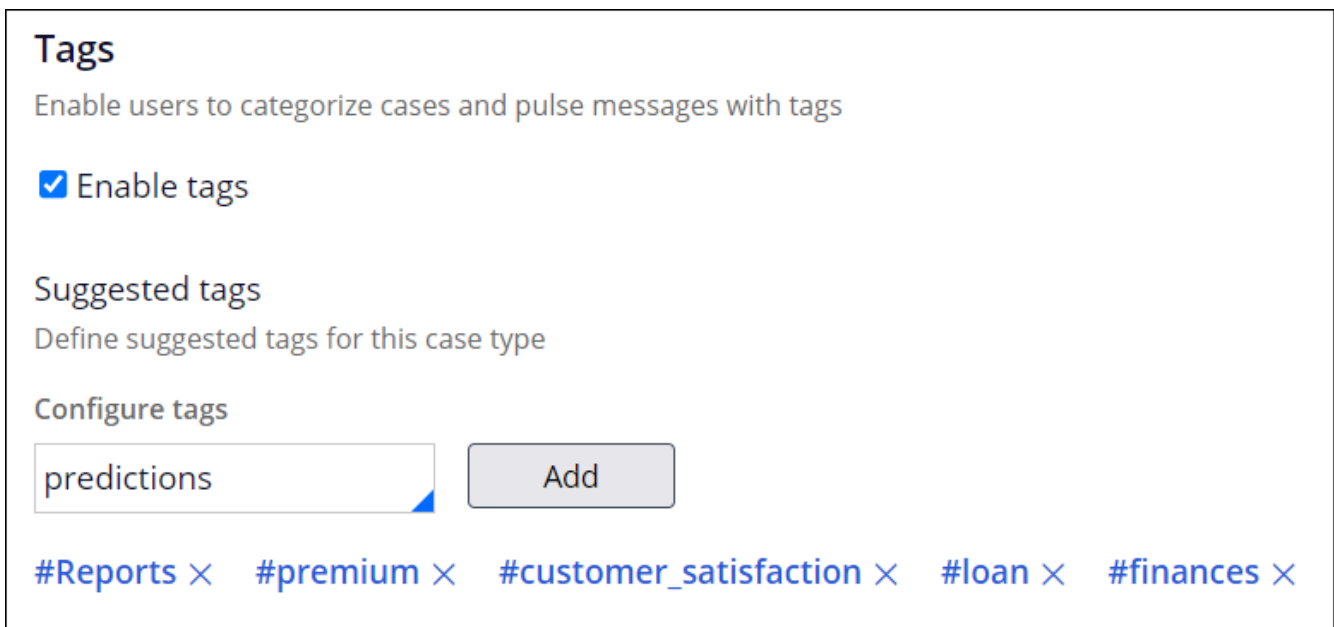




The tags widget showing tags that are associated with a case

**Note:** For applications that you build with the traditional Theme UI-Kit, see [Tagging a case](#).

You can create an initial list of tags for your application, to provide end users with tagging suggestions. The list helps end users find and apply the most often used tags and promotes consistent tagging throughout the application. For more information, see [Creating suggested tags for cases \(on page 488\)](#).



Enabling tags for a case type and creating the initial set of tags

At run time, you can tag cases by using existing tags or by adding new tags. You can tag a case regardless of the current case status. For more information, see [Adding tags to cases at run time \(on page\)](#)

## Manage tags

Ac

### Added tags

#premium ✕ #reports ✕ #predictions ✕ #invoice ✕ #finances ✕

### Suggested tags

#finances #invoice #Predictions #Reports #premium

### Tags recently used by me

#premium #Reports #Predictions #invoice #finances

selecting tags from different categories

You can gather relevant information faster by filtering tagged cases in several ways, for example, by recently added, recently viewed, or most

frequently used tags. For more information, see [Filtering tagged cases \(on page 487\)](#).

The screenshot shows the Pulse interface with the Tags widget on the left and a filtered list of cases on the right.

**Tags Widget:**

- Search bar: premium
- Recently added by me (3): #Reports, #premium, #finances
- Recently added by anyone (16): #Reports, #premium, #finances, #tag672, #tag671, #spacetag5, #spacetag2, #spacetag1, #tag1, #tag108, #tag2, #tag109, #spacextag, #doctag, #testdoc2, #testdoc1

**Selected tags:** #premium (Edit)

**CASES (2) POSTS (1)**

Group Fields Density Refresh

Name	Case type	ID	Status	Last Update	Priority
<a href="#">Loan application</a>	Loan application	L-370...	New	2 months 16 days a...	10
<a href="#">Loan application</a>	Loan application	L-240...	New	3 months ago	10

[Configuring the Tags widget \(on page 487\)](#)


[Tagging a message in Pulse \(on page 506\)](#)

[Collaborating with users by using Pulse \(on page 499\)](#)

## Adding tags to cases at run time

Find relevant information faster and enhance collaboration on cases by associating tags with cases.


For example, in an insurance application, you can add tags such as `premium` or `high_risk` to convey vital information about a case.


 **Note:** For applications that you build with the traditional Theme UI-Kit, see [Tagging a case](#).

At run time, you can add tags to a case by selecting from a list of available tags or by creating new tags. For more information about creating a set of default suggested tags for cases, see [Creating suggested tags for cases \(on page 488\)](#).

You can tag a case regardless of its current status.

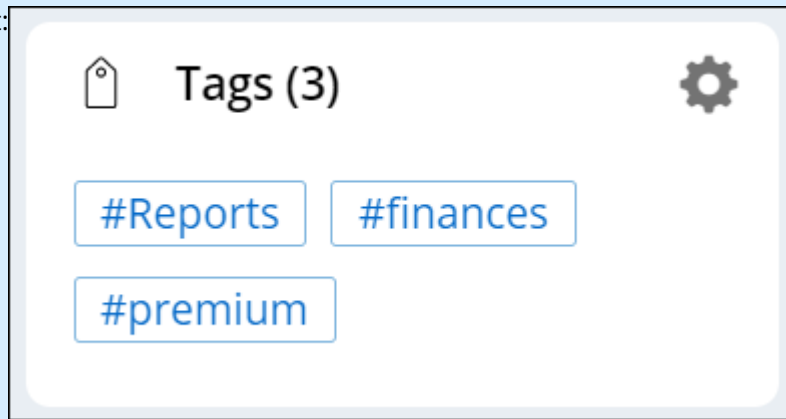
1. Log in to an end-user portal, for example User Portal.
2. Open a case:
  - Search for the case.
  - Click on the case in the worklist or work queue on your dashboard.
3. In the Summary pane of the case, click **Actions > Manage tags**.
4. In the **Manage Tags** dialog box, select a tag from the list of existing tags or enter a new tag name, for example `Reports`.

 **Note:** Tags cannot contain spaces and are limited to 64 characters.

 **Tip:** You can also add tags by clicking items in the **Tags recently used by me** section of the dialog box.

5. Click **Add**.
6. **Optional:** To add more tags, repeat steps [4 \(on page 486\)](#) to [5 \(on page 486\)](#).
7. **Optional:** To remove a tag association from a case, in the **Manage Tags** dialog box, click the **Unlink this tag** icon next to the tag that you want to remove.
8. Close the dialog box.

The tags that you add appears in the Tags widget, in the associated tags list:



List of tags associated with your case



**Note:** You can now manage case tags by clicking the **Manage tags** icon, and then adding or removing tags.

## Filtering tagged cases

Use tags to search, access, and report on cases faster. You can filter tagged cases in several ways, for example, by viewing only the recently added or most frequently used tags. To find specific information, you can also create specific tags and then search cases with those tags.

For example, you can look for cases with the `premium` tag to identify your key clients or accounts.



**Note:** For applications that you build with the traditional Theme UI-Kit, see [Tagging a case](#).

1. Log in to an end-user portal, for example, User Portal.
2. In the navigation pane on the left, click **Tags**.
3. Search for cases in one of the following ways:
  - To filter cases by a specific tag, enter the tag name in the **Search** field, and then select the tags from the **Search results** section.
  - To filter cases by tags that you added recently, select the tags from the **Recently added by me** section.

- To filter cases by tags that any other user added recently, select the tags from the **Recently added by anyone** section.
- To apply more out-of-the-box filters that categorize tags, click the **Filter** icon next to the **Search** field, and then select any of the available options, for example, **Recently viewed by me**.

The **Selected tags** section displays cases that use the selected tags. You can now preview the cases by clicking the case names.



**Note:** If your tags also appear in any Pulse messages, the search results provide this information on the **Posts** tab.

The screenshot shows the 'Tags' section in App Studio. A search box contains 'premium'. Below it, there are two sections: 'Recently added by me (3)' and 'Recently added by anyone (16)'. The 'Selected tags' section shows '#premium' with an 'Edit' button. Below that, there are two tabs: 'CASES (2)' and 'POSTS (1)'. The 'CASES (2)' tab is active, showing a table of cases filtered by the 'premium' tag.

Name	Case type	ID	Status	Last Update	Priority
Loan application	Loan application	L-370...	New	2 months 16 days a...	10
Loan application	Loan application	L-240...	New	3 months ago	10

Filtering cases by the premium tag

4. **Optional:** To add more tags to filter cases that use any of the selected tags, in the **Selected tags** section, click **Edit** to enter or select more tags in the search box, and then click **Apply**.

## Creating suggested tags for cases

Improve case search and ensure that users apply consistent tagging by creating suggested tags. When users apply consistent tagging in cases, they obtain information that they need faster. By defining sets of suggested tags you shorten case processing time as users can select tags from a list when they tag a case instead of typing tags manually.

1. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
3. Click the **Settings** tab, and then, in the navigation pane, click **Collaboration**.
4. In the **Tags** section, in the **Configure tags** field, enter a tag that you want to create as a suggested tag, and then click **Add**.

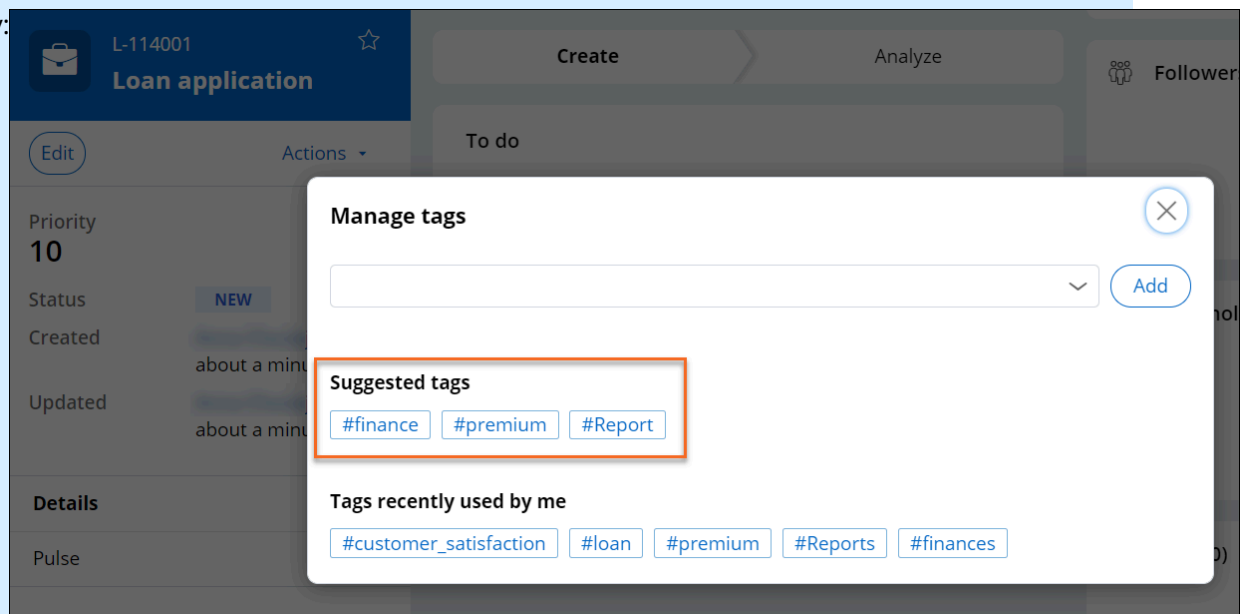


**Note:** Tags cannot contain spaces and are limited to 64 characters.

- ⓘ Tags can use underscores, so a tag with multiple words can be written as `customer_satisfaction`.

5. **Optional:** To remove a tag from the list of default tags associated with a case type, click **Remove** next to the tag.
6. Click **Save**.

At run time, when users tag a case, they can select the suggested tags, instead of entering tags manually:



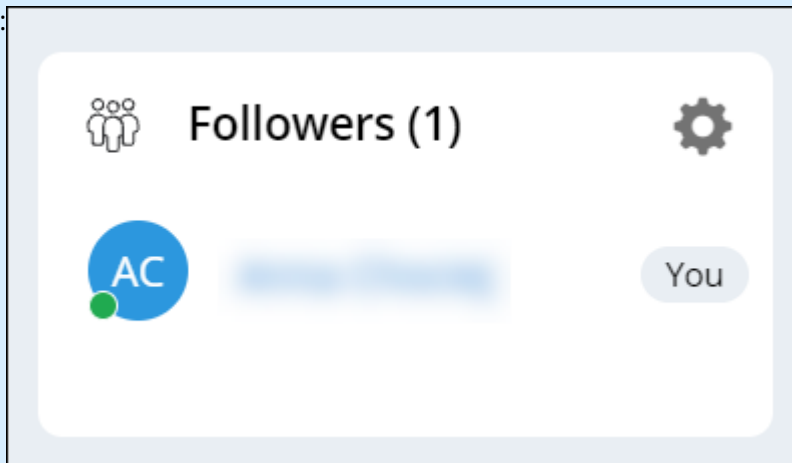
A case with a list of suggested tags

## Following a case

Follow a case to receive notifications when users post messages to the case in Pulse. By actively monitoring cases, you can contribute to stakeholder discussions and help resolve cases more quickly.

1. Log in to an end-user portal, for example, the User Portal.
2. Open a case:
  - Search for the case.
  - Click on the case in the worklist or work queue on your dashboard.
3. In the header of the case, click the **Star** icon.
4. **Optional:** To unfollow the case, click the **Star** icon again.

You now follow a case and the utilities pane displays you as a follower of the case:



Case followers in the utilities pane of a case

### Related information

Widgets in Cosmos React (on page )

## Managing case followers

Ensure that relevant stakeholders participate in case discussions by controlling which users follow a case.

1. Log in to an end-user portal, for example User Portal.
2. Open a case:
  - Search for the case.
  - Click on the case in the worklist or work queue on your dashboard.
3. In the **Followers** section, click the **Manage followers** icon.
4. In the **Followers** dialog box, in the drop-down list, select one or more users in your application, and then click **Add**.
5. **Optional:** To remove a user who is no longer a relevant stakeholder from the followers list, click the **Delete** icon next to the user.  
You can undo your changes by clicking the **Add** icon next to a deleted follower.
6. Close the **Followers** dialog box.

Case followers receive notifications for the case, based on the settings in their notification preferences.

## Best practices for managing interactions between parent and child cases

Understanding how to correctly handle parent and child cases is crucial to maintaining data consistency and ensuring accurate case processing, especially when interacting with parent and child cases on multiple tabs within the same user session.

**Note:** Applicable to Theme Cosmos (on page ) applications.

In Pega Platform™ Case Management, you can start multiple child cases from a parent case and work on the child cases in parallel to the parent case, which speeds up case resolution.

Typically, you use child cases when you want to model work:

- that is performed by different parties
- that follows different reporting options
- that follows different time frames
- that is common to multiple case types

You add a child case to a case type by adding a Create case automation step to the case life cycle. When encountering the Create case step, the system creates an instance of the specified child case type and copies data from the parent case (cover case), to the child case (covered case).

When a user works on both parent and child cases in separate tabs within the same session, it's essential to understand the expected behavior. If a child case is resolved, the parent case's data on the clipboard is not automatically updated. When the user resolves the child case and goes back to the parent case, the child case data is not updated on the parent case clipboard unless you specifically refresh the parent case. To prevent committing stale data, you must refresh the parent case tab before continuing with its processing.

---

### Related information

[Creating a child case type \(on page 31\)](#)

## Creating tools for facilitated work processing

Pega Platform™ offers a variety of tools that give you a quick access to key metrics about cases and teams, so that you can manage your teams and work more effectively.

## Configuring teams

To ensure that you engage all the people necessary to deliver your projects, configure a team, so that you can assign different roles to team members, and then manage work in a convenient and transparent way.

---

### Related information

[Inviting collaborators to your application \(on page \)](#)

[Collaborating on cases \(on page 499\)](#)



[Creating and managing cases \(on page 463\)](#)

[Skill rules \(on page \)](#)

## Creating a team

To increase productivity and facilitate distribution of the workload in your application development process, create a team. You can then assign work to team members, and speed up resolution of your cases by providing the team with collaboration tools.

You can also refer to teams as work groups.

1. In the header of Dev Studio, click **Launch portal**, and then select a portal that contains the **Teams** widget.
2. In the header of App Studio, navigate to a portal that contains the **Teams** widget.
3. In the navigation pane, click **Teams**.
4. In the header of the **My Teams** section, click **Create team**.
5. In the **Create team** window, provide information about the team:
  - a. In the **Name** field, enter a unique team name.
  - b. **Optional:** To provide more information about the team, in the **About** field, enter some text that describes the purpose of the team.
  - c. In the **Manager** field, press the Down arrow key, and then select the name of the user who supervises the work of the team.
  - d. Click **Submit**.
6. Add team members:
  - a. In the **My Teams** section, click the team name.
  - b. On the **Team** page, in the **Members** section, click the **Edit members** icon.
  - c. In the **Edit members** window, in the text field, press the Down arrow key, and then select a user name.
  - d. Click **Add**.
  - e. Click **Submit**.
7. **Optional:** To edit information about the team, on the **Team** page, click **Actions > Edit team information**.

## Adding work queues to a team

To speed up case resolution, improve workload management in your application by adding work queues to your team. You can create multiple work queues that collect tasks for users of different areas of expertise, for example, a work queue that lists tasks for managers.

When you create a new team, your application creates a default work queue for the team.



1. In the header of App Studio, click **Preview** to navigate to the portal Dashboard that contains the **Teams** widget.
2. In the **Teams** section, click on the name of the team to which you want to add a work queue.
3. In the **Work queues** section, click **Add new**.
4. In the text field that appears, define a work queue:
  - To create a new work queue, enter the name of the work queue.
  - To add an existing work queue, press the Down arrow key, and then select a work queue.
5. **Optional:** To add more work queues, repeat steps 3 (*on page 493*) and 4 (*on page 493*).

## Adding a secondary manager to a team

Increase the visibility of your work by associating your team with more than one manager.

By default, each team has one manager who supervises the work of the team and can assign, transfer, or work on assignments in a case.

1. In the navigation panel, click **Records > Organization > Work Group**.
2. In the **Work Group Name** column, click the name of a team.
3. If no operator IDs are listed in the **Authorized Managers** section, click **Add item**.
4. In the **Manager ID** field, press the Down Arrow key, and then select an operator ID.
5. Click **Save**.

The secondary manager can access the team's work queue but cannot complete assignments or receive requests for approval.

## Defining areas of expertise for a team

As a team manager, define areas of expertise for your team to ensure that your application routes assignments to users with relevant skills.

1. In the Case Manager portal, click **My Teams**.
2. As the application author or case manager, click **My Teams**.
3. In the **Members** section, click the avatar of a team member.
4. Click **Actions > Edit profile**.
5. Click **New skill**.
6. In the **Skill** field, press the Down arrow key, and then select a skill that is pertinent to the role and title of the team member.
7. In the **Proficiency** field, enter an integer from 0 to 10 that rates how competent the team member is when applying this skill to the assignments in your application.
8. Click **Submit**

## Deleting a team

Delete a team when it does not have any associated users or work queues to reduce complexity in your application.

1. In the header of Dev Studio, click **Launch portal**, and then select a portal that contains the **Teams** widget.
2. In the header of App Studio, navigate to a portal that contains the **Teams** widget.
3. In the navigation pane, click **Teams**, and then click a team name.
4. On the **Team** page, click **Actions > Delete team**.
5. In the **Delete Team** dialog box, enter the reason why the team is no longer required.
6. Click **Submit**.

## Dashboard management

Increase the productivity of your team by configuring a dashboard that consolidates and presents relevant data in the most convenient form. For example, you can design a workspace that includes several reports that help managers oversee work by displaying case status, sales data, and burn down report widgets.

At run time, end users can personalize their dashboard instance to display information according to their specific preferences and interact with data in the most convenient way.



**Note:** Dashboards are available in applications that use Theme UI-Kit, Theme Cosmos, and Cosmos React.

For applications that you build with the traditional Theme UI-Kit, see [Configuring your dashboard](#).

In Theme Cosmos applications, you can use two types of dashboard components:

- The traditional dashboard component that uses a section-based My Dashboard page, which is an out-of-the-box custom page.
- The new Cosmos React dashboard component, which is a React-based landing page with the dashboard template. For more information, see [Configuring a dashboard \(on page 494\)](#).

In Cosmos applications, you can create a React-based landing page that uses the dashboard template. The Cosmos React dashboard component is a curated experience. Consequently, the application designers specify what users see when they log in. Personalization at run time is not possible. For more information, see [Configuring a dashboard \(on page 494\)](#).

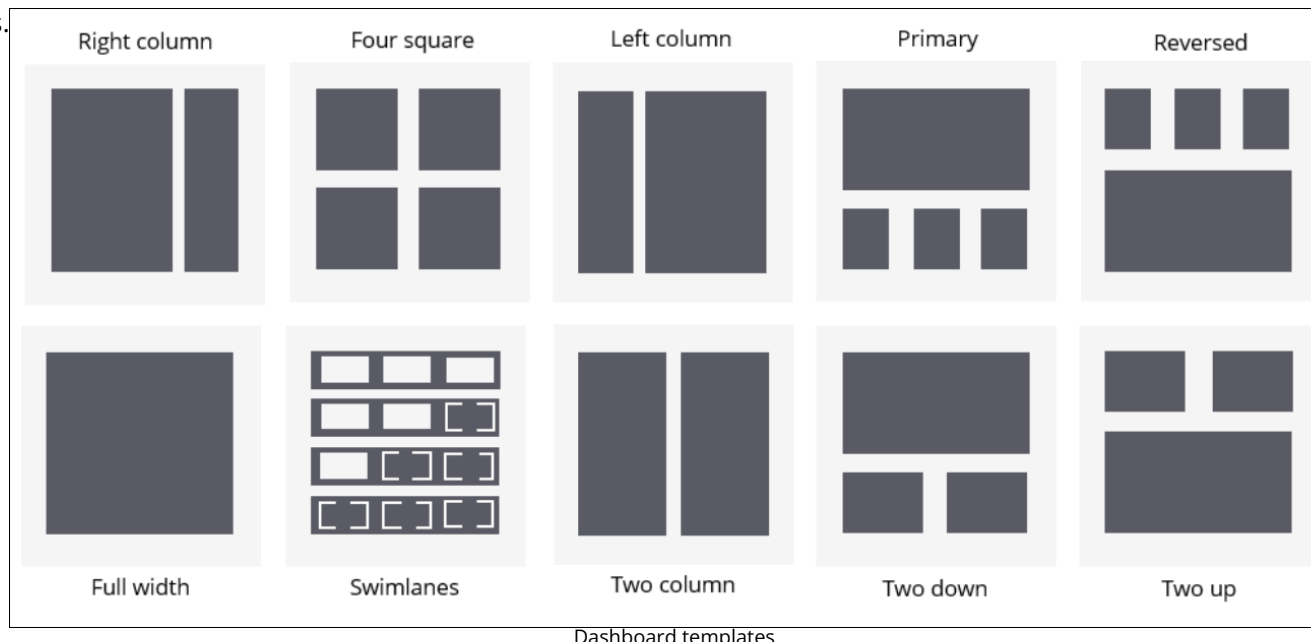
## Traditional dashboard component

By default, Pega Platform™ uses role-based interfaces, which includes a dashboard tailored to a role. For example, when a user with the Manager role signs in to a Pega Platform application, the user is presented with a dashboard that is tailored to include information necessary to manage the users who complete the work.

The developer can configure the dashboard, which is available as a custom page in an application, for a role by publishing changes to the default dashboard. For example, the manager might see the number of open cases and overdue assignments in order to better manage the workload of their direct reports. End users can then access the published default dashboard that is assigned to their role.

End users can personalize their dashboard by switching templates and adding or removing widgets from slots. A template defines the layout of a dashboard and divides the workspace into slots. Slots are containers for one or more widgets, which are elements of the user interface that group summary information and Key Performance Indicators (KPIs) from different sources and present them in an actionable format. If end users personalize a dashboard, changes that developers publish to a default dashboard do not overwrite or reset the personalized dashboard of the end users.

Pega Platform provides various templates that allow developers and end users to customize their dashboard, including which widgets to view in which slots. The provided templates have different numbers with slots in different orientations.



For relevant training materials, see the [Customizing a dashboard](#) challenge on Pega Academy.

Hybrid mode (on page )

Cosmos React (on page )



[Creating a team \(on page 492\)](#)

[Collaborating on cases \(on page 499\)](#)

[Summary panel views \(on page \)](#)

## Changing the dashboard template

Find basic information that you need more efficiently by organizing your dashboard layout in a comprehensive and transparent way. Dashboards that offer quick access to information that is necessary for a particular role can help increase productivity. For example, in sales operations, the manager role needs a more comprehensive dashboard to accommodate more data tracking tools than a sales representative, who might benefit from trimming down a too-elaborate dashboard.

1. Log in to an end-user portal, for example, the User Portal.
2. In the navigation menu of your application, click **Dashboard**.
3. In the upper-right corner of your workspace, click the **Personalize the dashboard** icon.
4. In the **Edit dashboard** pane, click **Switch template**.
5. In the **Select a template** modal dialog box, select the template that you want to use.
6. Click **Publish**.

## Adding a widget to your dashboard

Increase the productivity of your team by personalizing the dashboard with widgets that display relevant information, such as charts or shortcuts to reports and frequently used resources. For example, you can design a workspace that includes a widget for frequently used reports to help managers save time and oversee their projects.

1. In the navigation menu of your application, click **Dashboard**.
2. In the upper-right corner of the work space, click the **Personalize the dashboard** icon.
3. In the **Edit dashboard** pane, choose a widget:
  - a. In the slot where you want to position the widget, click **Add widget(s)**.
  - b. In the **Add widgets to slot** modal dialog box, select one or more widgets that you want to add to the slot, and then click **Add selected**.

The new widget appears at the bottom of the slot.

- c. **Optional:** To change the position of the widget in the work area, drag the widget into a new position or a different slot by following the guide markers that are displayed on the screen.
- d. **Optional:** To remove any of the widgets, click the **Delete this widget** icon.





4. In the work area, hover over the newly added widget, and then determine whether there are additional options to configure:

Choices	Actions
Click to edit message on hover	<ul style="list-style-type: none"> <li>a. Click the widget.</li> <li>b. In the <b>Edit dashboard</b> pane, edit the widget with the configuration controls.</li> <li>c. Click <b>Save</b>.</li> <li>d. Click <b>Publish</b>.</li> </ul>
No options message on hover	Click <b>Publish</b> .

## Dashboard widgets

A dashboard widget displays operational information about your application and key performance indicators, such as the number of assigned items or the average time to resolve a case. By personalizing your dashboard with widgets that are relevant to your role, you can increase productivity.

The following table lists the widgets that you can add to your dashboard in a Theme Cosmos application:

Widget	Description	Configurable
<b>Case lifecycle</b>	Displays the stages and steps for each case type in your application.	No
<b>Case Manager shortcuts</b>	Provides shortcuts to the <b>Reports</b> and <b>Get next work</b> items.	No
<b>Cases entered by me</b>	Displays all cases that you create.	No
<b>Case status</b>	Renders a pie chart that breaks down cases by stage.	No
<b>Connected Apps</b>	Displays a list of third-party apps and their connection status. You can also connect and disconnect apps from a service provider.	No
<b>Following</b>	Displays a responsive list of work items that you are currently following.  Options include changing the number of items displayed per page and the visibility of the <b>Bulk unfollow</b> option.	Yes
<b>My Cases</b>	Displays all open cases to which you are a party.	Yes

Widget	Description	Configurable
	Options include changing the title of the widget and the visibility of the <b>Quick Create</b> option.	
<b>Nearby Cases</b>	Displays cases within a maximum radius of 50 miles from your current location.  Options include selecting the case type and the <i>Location</i> property.	Yes
<b>Recents</b>	Displays a list of cases that you recently updated or opened.	No
<b>Team members</b>	Displays the users in your work group and the number of their open assignments.	No
<b>Teams</b>	Displays a list of teams within your work group.	No
<b>Work queues</b>	Displays a list of work queues in your application.	No
<b>Worklist</b>	Displays a list of work items that are assigned to you.  Options include changing the source of the list to a specific work queue or work group.	Yes
<b>Data Imports - In-progress</b>	Displays the details of currently running data imports.	No
<b>Data Imports - Recently completed</b>	Displays the details of recently completed data imports.	No
<b>Achievement Tracker</b>	Displays badges that you receive by accomplishing specific tasks in the product.	No
<b>Google Maps</b>	Displays a map that shows your current location and open assignments.  Options include selecting an automatic or fixed height for the map.	Yes
<b>Welcome to Pega Express</b>	Displays introductory information about the product and an overview video.	No
<b>Welcome to Case Manager</b>	Displays introductory information about the portal and an overview video.	No
<b>Manage Change</b>	Displays data types that are delegated to you by a manager.	Yes

Widget	Description	Configurable
	<p>Displays the records delegated to the user's access group. Depending on their permissions, users can modify data types, decision trees, and paragraphs.</p> <p>Options include setting the visibility of the widget title.</p>	
<b>Recent reports</b>	<p>Displays a list of reports that you ran recently.</p> <p>Click a report title to view the result data.</p>	No
<b>Report widget</b>	<p>Displays the results of a specific report.</p> <p>Options include setting a report category and name.</p>	Yes
<b>Pulse Activity Feed</b>	<p>Displays the Pulse activity gadget that displays Pulse posts made by the user.</p>	No

### Collaborating on cases

Complete your Microjourney faster by providing collaboration tools for your application users, such as customer service representatives (CSRs). Through a transparent exchange of messages in an open discussion, CSRs can resolve cases faster and with better effect.

Application users can collaborate by posting Pulse messages, maintaining spaces, and working on documents.

### Collaborating with users by using Pulse

Resolve your work faster by using Pulse. By sharing information in Pulse, users can work together to complete their work more effectively.

---

#### Related information

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

### Posting a message in Pulse

Post a message in Pulse to share information, for example, the status of a case, or to ask a question in a conversation thread. By posting your message in the correct context, you control who can receive notifications, view your messages, and post replies.



For example, users who are preparing a presentation for a customer can exchange ideas in a Pulse conversation, which helps keep the discussion within the context of a single user story.

1. Choose the context for your message, based on your target audience and message content:

Choices	Actions
<b>Application developers and stakeholders</b>	In the header of App Studio, click the <b>Toggle Developer collaboration</b> icon.
<b>Case followers and participants</b>	a. Open a case by searching for it or by reviewing the worklists and work queues on your dashboard.
<b>Members of a team</b>	a. In the navigation pane of the Case Manager portal, click <b>My Teams</b> . b. Click the name of the team to which you want to address your message.
<b>A specific user</b>	a. In the navigation pane of the Case Manager portal, click <b>My Teams</b> . b. Click a team name. c. In the <b>Members</b> section, click <b>Edit Members</b> . d. In the <b>My team</b> list, click the name of the user to whom you want to address your message.

2. Add a message:

- To add a message for all users within this context, click **Post > Post**.
- To add a message for specific users within this context, click **Post > Private post**, press the Down arrow key, and then select the message recipients.



**Note:** All the users involved in a private conversation can add more users to the conversation by referencing them, but only the owner can remove users from the conversation.



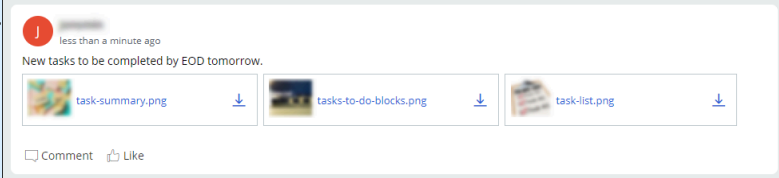
**Note:** Only standard Pega Platform applications support sending private Pulse messages. In applications based on Cosmos React, you can send only public messages.

- To create and assign a task in Pulse, click **Post > Task**.

For more information, see [Creating a task in Pulse \(on page 510\)](#).



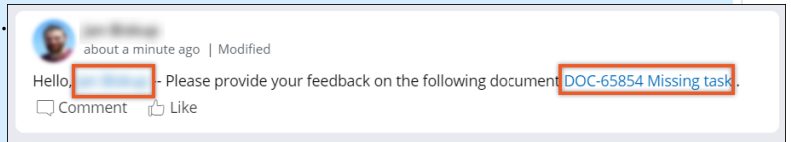
3. Click **Start a conversation**, and then enter your message text.
4. **Optional:** Enhance your message by controlling its presentation and the information that supports it:

Choices	Actions
<b>Personalize the look and feel of your message</b>	<ol style="list-style-type: none"> <li>a. Click the <b>View formatting help</b> icon.</li> <li>b. Apply a syntax from the <b>Style, Lists, or Headings</b> section to the relevant text in your message.</li> </ol>
<b>Add links</b>	<ol style="list-style-type: none"> <li>a. Click the <b>View formatting help</b> icon.</li> <li>b. Apply a syntax from the <b>Images or Links</b> section to the URL in your message text.</li> </ol>
<b>Provide additional information by adding an attachment</b>	<ol style="list-style-type: none"> <li>a. Click the <b>Add attachment</b> icon.</li> <li>b. Upload a file from your local system or an external location.           <div data-bbox="587 873 1463 1239" style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p>The system associates the attachment with your message and the context in which you post the message, for example, a case.</p>  <p style="text-align: center;">Attachments embedded in a Pulse message</p> </div> </li> </ol>
<b>Reference a user, space, document, or case</b>	<div data-bbox="509 1314 1463 1499" style="border: 1px solid #0070C0; padding: 10px; margin-bottom: 10px;"> <p><b>Note:</b> Only standard Pega Platform applications fully support this feature. Applications that you build on Cosmos React support referencing only users.</p> </div> <ol style="list-style-type: none"> <li>a. Place your cursor in the message text, and then enter @.</li> <li>b. Select whether you want to reference a user, space, document, or case.</li> <li>c. Type one or more characters to narrow the list of results, and then select a value.</li> </ol>

## Choices

## Actions

Individual users and users of the spaces that you reference receive notifications about your message, based on their notification settings. Documents and cases that you reference become links in your message.



Referencing a user and a document in a Pulse message

- d. **Optional:** To help users create messages more quickly, limit the list of cases that they can reference in a message. For more information, see [Restricting case references in Pulse \(on page 506\)](#).

## Add tags to your message



**Note:** Only standard Pega Platform applications fully support this feature. In applications that you build on Cosmos React, you can also create tags, but clicking a tag does not open a **Tags** landing page.

- a. Place the cursor in your message text, and then enter #.
- b. Type one or more characters, and then select a tag from the list of matching results.

If there are no matches, the application creates a new tag from the text that you enter.

- c. **Optional:** To add another tag, enter a blank space, and then repeat step 4.b (on page 502).

5. Click **Post**.

[Replying to a message in Pulse \(on page 503\)](#)

[Extension points and supporting rules for attachments \(on page 461\)](#)

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

## Replying to a message in Pulse

You can reply to a message in Pulse to continue a conversation. For example, a project manager responds to a message in which one of the developers asks for clarifications about a task in the upcoming sprint.

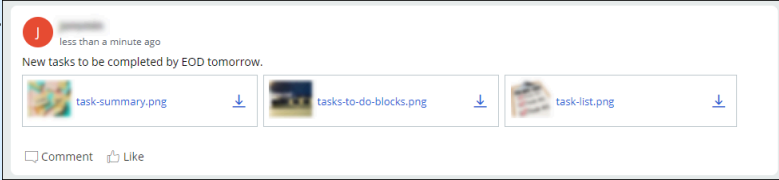
When you reply to another user's reply, the application automatically adds the name of the user at the beginning of your comment.

1. Choose the context for your message, based on your target audience and message content:

Choices	Actions
<b>Application developers and stakeholders</b>	In the header of App Studio, click the <b>Toggle Developer collaboration</b> icon.
<b>Case followers and participants</b>	a. Open a case by searching for it or by reviewing the worklists and work queues on your dashboard.
<b>Members of a team</b>	a. In the navigation pane of the Case Manager portal, click <b>My Teams</b> . b. Click the name of the team to which you want to address your message.
<b>A specific user</b>	a. In the navigation pane of the Case Manager portal, click <b>My Teams</b> . b. Click a team name. c. In the <b>Members</b> section, click <b>Edit Members</b> . d. In the <b>My team</b> list, click the name of the user to whom you want to address your message.

2. Find the relevant message in the Pulse conversation.
3. **Optional:** To acknowledge the message without sending a reply, click the **Like** icon.
4. Click **Comment**, and then enter your message text.
5. **Optional:** Enhance your message by controlling its presentation and the information that supports it:

Choices	Actions
<b>Personalize the look and feel of your message</b>	a. Click the <b>View formatting help</b> icon.  b. Apply a syntax from the <b>Style, Lists, or Headings</b> section to the relevant text in your message.

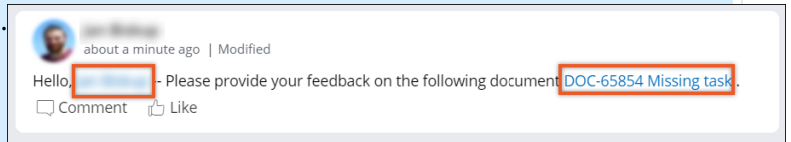
Choices	Actions
<p><b>Add links</b></p> <p><b>Provide additional information by adding an attachment</b></p>	<p>a. Click the <b>View formatting help</b> icon.</p> <p>b. Apply a syntax from the <b>Images</b> or <b>Links</b> section to the URL in your message text.</p> <hr/> <p>a. Click the <b>Add attachment</b> icon.</p> <p>b. Upload a file from your local system or an external location.</p> <div data-bbox="589 537 1463 898" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p>The system associates the attachment with your message and the context in which you post the message, for example, a case.</p>  <p style="text-align: center;">Attachments embedded in a Pulse message</p> </div>
<p><b>Reference a user, space, document, or case</b></p>	<div data-bbox="509 978 1463 1163" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p><b>Note:</b> Only standard Pega Platform applications fully support this feature. Applications that you build on Cosmos React support referencing only users.</p> </div> <p>a. Place your cursor in the message text, and then enter @.</p> <p>b. Select whether you want to reference a user, space, document, or case.</p> <p>c. Type one or more characters to narrow the list of results, and then select a value.</p> <div data-bbox="589 1514 1463 1661" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p>Individual users and users of the spaces that you reference receive notifications about your message, based on their notification settings. Documents and</p> </div>



## Choices

## Actions

cases that you reference become links in your message.



Referencing a user and a document in a Pulse message

- d. **Optional:** To help users create messages more quickly, limit the list of cases that they can reference in a message. For more information, see [Restricting case references in Pulse \(on page 506\)](#).

## Add tags to your message



**Note:** Only standard Pega Platform applications fully support this feature. In applications that you build on Cosmos React, you can also create tags, but clicking a tag does not open a **Tags** landing page.

- a. Place the cursor in your message text, and then enter #.
- b. Type one or more characters, and then select a tag from the list of matching results.

If there are no matches, the application creates a new tag from the text that you enter.

- c. **Optional:** To add another tag, enter a blank space, and then repeat step 5.b (on page 505).

6. Click **Post**.

Users receive notifications about your reply and any attachments that you provide, based on their notification preferences.

[Setting notification preferences \(on page 346\)](#)

[Posting a message in Pulse \(on page 499\)](#)

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)



## Restricting case references in Pulse

You can control which cases users can reference in the messages that they post to Pulse. By limiting the choices in the list of suggested cases, you can help users create messages more quickly.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Reports** category, and then click **Report Definition**.
3. Open the *pyGetCasesForReferencesFromIndex* report definition that applies to the *Work-* class.
4. Copy the report definition to an open ruleset in your application. For more information, see [Copying a rule or data instance \(on page 499\)](#).
5. Create a filter condition that excludes cases or instances of a specific case type.
  - a. On the **Query** tab, in the **Edit filters** section, click **Add filter**.
 

Your application populates the **Condition** field and adds its value to the Boolean expression in the **Filter conditions to apply** field.
  - b. In the **Column source** field, enter `.pxObjClass`.
  - c. In the **Relationship** list, select **Is not equal**.
  - d. In the **Value** field, enter the class of the case type to exclude.
  - e. **Optional:** To exclude more case types, click **Select values**.
6. Click **Save**.

---

[Posting a message in Pulse \(on page 499\)](#)

[Finding rules by class \(on page 499\)](#)

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

## Tagging a message in Pulse

Categorize a message by tagging it in Pulse. By associating similar messages, you can help users find information that they need to resolve a case. For example, if all Pulse comments that are related to a bug contain the bug ID tag, users who work on the bug can quickly find all of the related information.



**Note:** Only standard Pega Platform applications support adding tags to existing messages. Applications that you build on Cosmos React do not support this feature.

The following guidance applies to case messages, however, you can communicate in other contexts, such as a team or application. For more information, see [Posting a message in Pulse \(on page 499\)](#).

1. Open a case:
  - Search for the case.
  - Click on the case in the worklist or work queue on your dashboard.
2. Hover over a message.
3. Click the **Show more** icon, and then click **Add tag**.
4. Add a tag to the message:
  - To create a new tag, enter a unique name in the text field.
  - To reuse a tag, press the Down arrow key in the text field, and then select a tag.
5. Click **Add**.
6. Click **OK**.

---

Case tags (on page )

[Posting a message in Pulse \(on page 499\)](#)

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

## Managing Pulse messages

Actively manage messages in Pulse so that you can quickly find relevant conversations or attachments.

Use the following techniques to manage your messages in Pulse:

---

[Collaboration with users by using Spaces \(on page 510\)](#)

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

## Viewing your activity feed

You can view your activity feed to find Pulse messages that you post or bookmark, and messages that are related to you. By filtering information from an aggregate source, you can find a specific conversation without opening individual cases.

1. In the navigation panel, click **Pulse**.
2. **Optional:** To find a message more quickly, filter the results.
  - a. Click the **Filter** icon.
  - b. Clear the check box next to a type of message to exclude it from the results.
  - c. Click **Apply**.
3. Review the list of results to find your message.
4. **Optional:** To continue the conversation for this message, click **Comment**.



---

[Adding the Pulse gadget to your application \(on page 362\)](#)

[Replying to a message in Pulse \(on page 503\)](#)

## Bookmarking a message in Pulse

You can bookmark a message in Pulse that you consider important, so that you can easily find the message later in your activity feed.

The following guidance applies to case messages, however, you can communicate in other contexts, such as a team or application. For more information, see [Posting a message in Pulse \(on page 499\)](#).

1. Open a case:
  - Search for the case.
  - Click on the case in the worklist or work queue on your dashboard.
2. Hover over the message, and then click the **Show more** icon.
3. Click **Bookmark**.

---

[Removing a bookmark in Pulse \(on page 508\)](#)

[Posting a message in Pulse \(on page 499\)](#)

[Viewing your activity feed \(on page 507\)](#)

## Removing a bookmark in Pulse

You can remove a bookmark in Pulse when a message is no longer relevant to you.

1. In the navigation panel, click **Pulse** to open your activity feed.
2. Click the **Filter** icon.
3. Select the **My bookmarked messages** check box.
4. Click **Apply**.
5. Hover over a message, and then click the **Show more** icon.
6. Click **Remove Bookmark**.

---

[Bookmarking a message in Pulse \(on page 508\)](#)

[Posting a message in Pulse \(on page 499\)](#)

[Viewing your activity feed \(on page 507\)](#)

## Following messages from a specific user

Follow users to view the posts that they make in your Pulse activity feed.




1. Open the user profile for the user that you want to follow.  
For example, you can open the profile of a team member by completing the following steps.
  - a. Click **My Teams**.
  - b. Click a team name.
  - c. In the **Members** section, click the name of the user.
2. Click **Follow**.

Your user profile displays the users whom you follow and the users who are following you. An icon indicates whether a user is online.

[Collaboration with users by using Spaces \(on page 510\)](#)

## Unfollowing messages from a specific user

Unfollow users so that you no longer see their posts in your Pulse activity feed.

 **Tip:** If you have the user's profile open, you can quickly unfollow the user by clicking **Following**.

1. From your Profile menu, select **Profile**.
2. In the **Following** section, click **See all**.
3. Click the **Check box** icon for each user that you want to unfollow.
4. Click **Submit**.

[Following messages from a specific user \(on page 508\)](#)

## Deleting a message in Pulse

You can delete a message in Pulse when it is no longer relevant to a conversation.

1. [Find the message in your activity feed \(on page 507\)](#).
2. Hover over the message, and then click the **Show more** icon.
3. Click **Delete**.

[Posting a message in Pulse \(on page 499\)](#)

[Replying to a message in Pulse \(on page 503\)](#)

## Creating a task in Pulse

To manage case work that needs to be completed within a specific period of time, create a task in Pulse and assign the task to other case users or to yourself. For example, you can assign a task to yourself as a reminder to upload address documents to a Car Loans case by the end of the day.

1. Open a case:
  - Search for the case.
  - Click on the case in the worklist or work queue on your dashboard.
2. Click **Post > Task**.
3. Enter a name for the task.
4. In the **Assignee** field, press the Down Arrow key, and then select a user to assign the task.
5. In the **Due date** field, select the date and time by which the task should be completed.
6. **Optional:** In the **Additional details** box, enter more information about the task.
7. Click **Create**.

Your Pulse interface and the Pulse interface of the assignee display the task. If the assignee does not complete the task, they receive two reminder notifications based on their notification preferences. The first notification arrives when half of the time till the due date elapses, and the second one appears after the deadline.

[Posting a message in Pulse \(on page 499\)](#)

[Preparing for collaboration with users by using Pulse \(on page 360\)](#)

## Collaboration with users by using Spaces

By creating spaces, you gather users that are professionally connected within a single digital community, so that they can collaborate on specific areas of interest. When you restrict the discussion of certain topics to the users of a space, you avoid broadcasting irrelevant messages to all users in your application and create more effective communication.

For example, you are a team leader who wants to hire new members for your team. To discuss prospective candidates with management, you can create a Hiring space that includes the manager, senior manager, and director of the team.



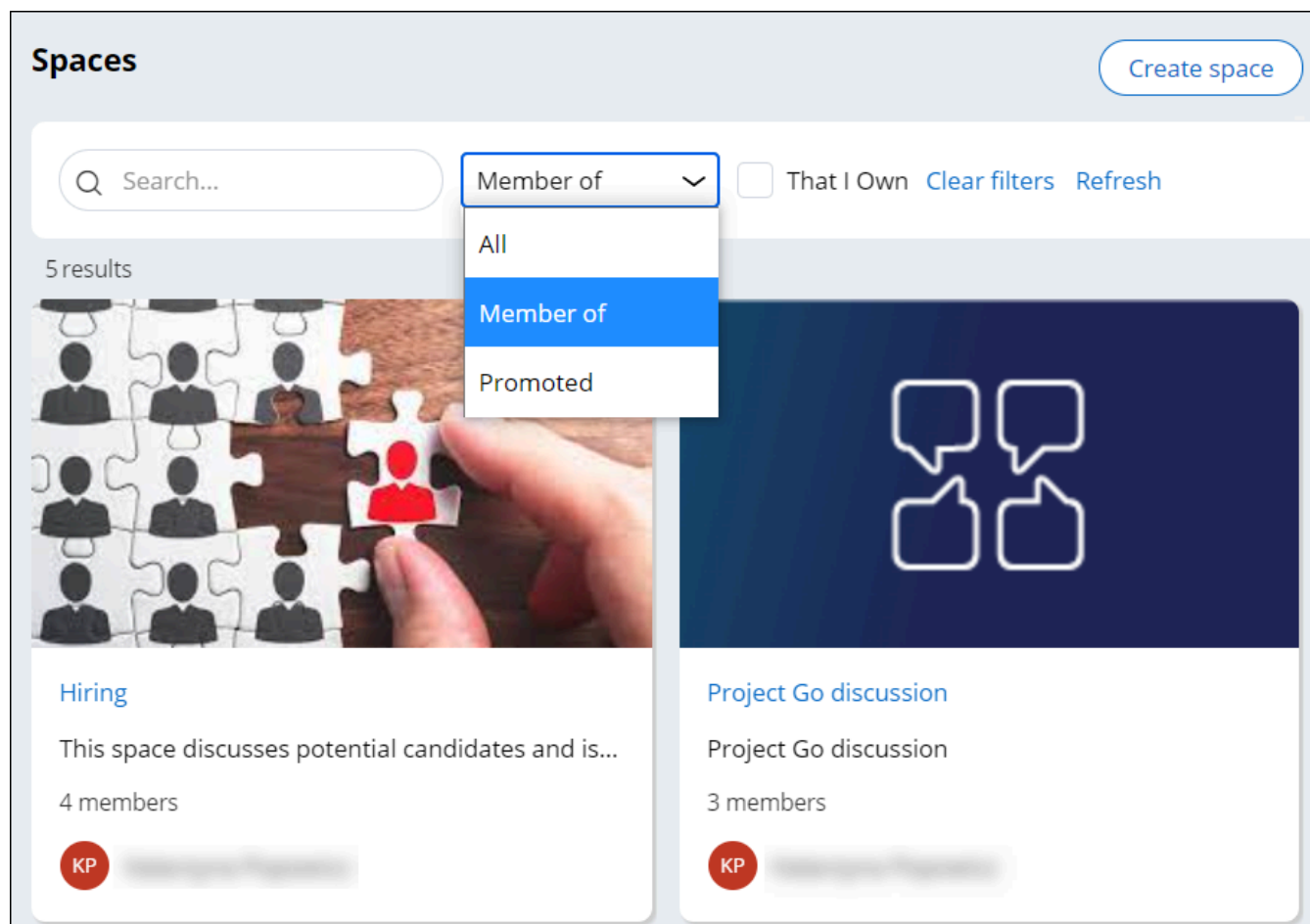
**Note:** The feature is not available in Cosmos React applications. You can only configure an application to launch the feature in a section-based UI in a new browser tab. For more information, see [Adding Theme Cosmos landing pages to the navigation menu \(on page \)](#) and [Launching Theme Cosmos pages from Cosmos React portals \(on page \)](#).

Spaces are areas in the application that enable groups of users to collaborate on a specific area of interest. In spaces, users can collaborate and exchange information by posting Pulse messages, uploading attachments, linking documents, or by pinning cases, documents, reports, and other spaces.

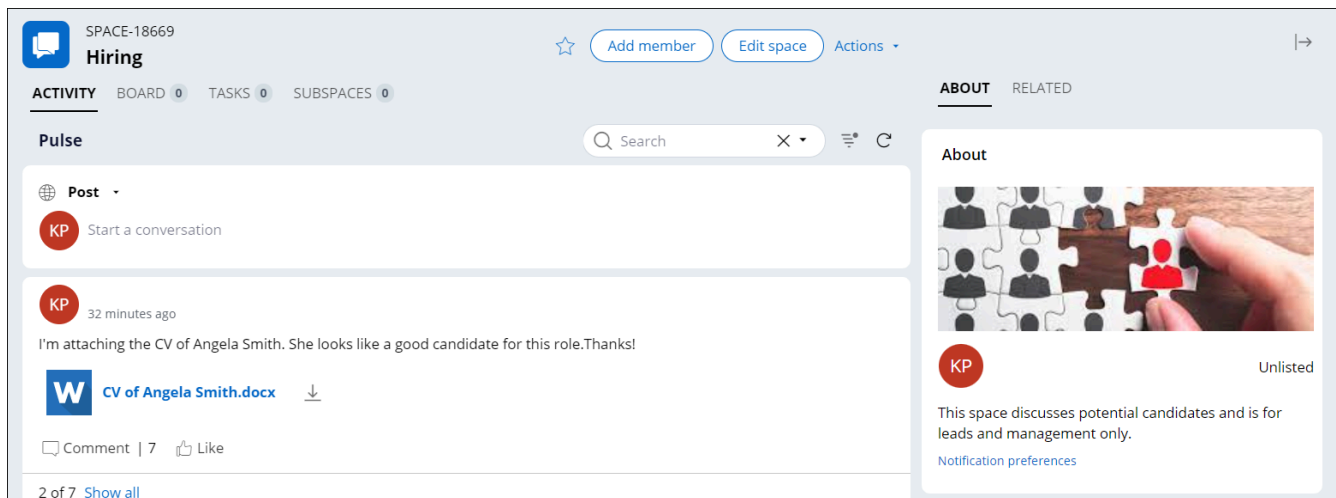
To distribute work, plan outcomes, and ensure that every space member is on track, you can create a task board, and then manage tasks in a comprehensible and clear way, even if the members of a space belong to different teams. For instance, you can populate a task board that is visible to managers and members of various teams that cooperate on a joint project.

To make your content more relevant, create a sub-space that gathers users within a space. For example, you can create a sub-space for CSRs that work together on a particular project, so that you do not involve the whole team.

The following images show the Spaces landing page and the collaboration interface for a space:



The Spaces landing page by default displays the spaces of which the user is a member



The collaboration interface for a space

The following tasks can help you share information with other users by using spaces:

[Collaborating with users by using Pulse \(on page 499\)](#)

[My Pega: My Spaces \(video\)](#)

[Spaces overview \(video\)](#)

## Creating a space

Create a space to collaborate with users in your application on a specific area of interest, for example, to discuss training for new hires in your team.

1. In the header of App Studio, navigate to a portal that contains the **Spaces** widget.
2. In the navigation pane, click **Spaces**.
3. Click **Create space**.

**Note:** The user who creates a space becomes the owner of the space.

4. Enter a name for the space.
5. **Optional:** To give a short description to the space, enter some relevant text in the **Description** field.
6. In the **Space type** menu, select who can see and access the space.
  - **Public** – All users in your application can see and join the space. The owner can also invite users to the space.
  - **Private** – All users in your application can see the space and request access. The owner can also invite users to the space.
  - **Unlisted** – Only the owner and users who are invited by the owner to join the space can see it.



7. **Optional:** To create tasks in Pulse and use a task board in your space, select the **Enable task tracking** checkbox.

You can also enable task tracking after you create a space by clicking **Edit space** on the **Activity** tab of your space.

For more information, see [Creating and managing tasks in a space \(on page 522\)](#) and [Creating a task in Pulse \(on page 510\)](#).

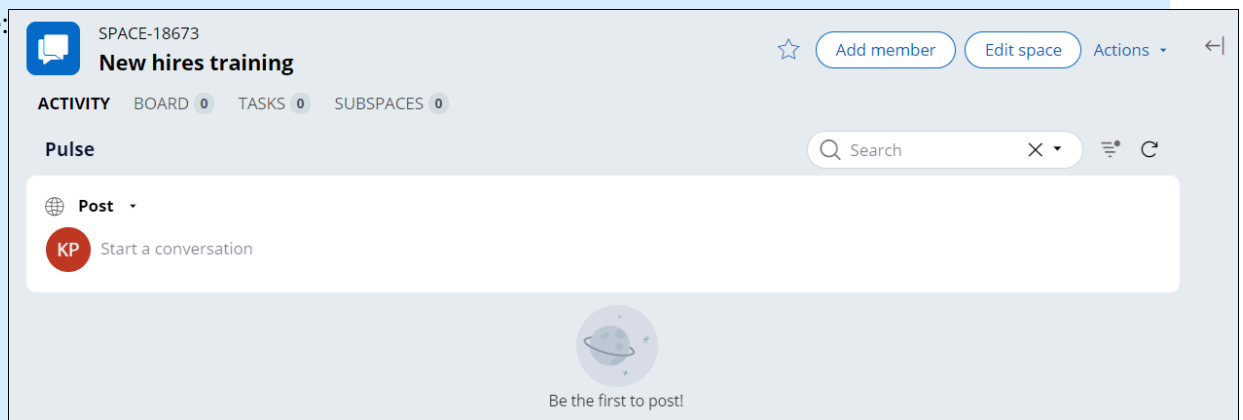
8. **Optional:** To associate an image with the space, in the **Featured image** section, click **Update space image**, and then browse to and upload an image.

The image is visible in the space preview on the Spaces landing page and when you open the space.

9. Click **Done**.

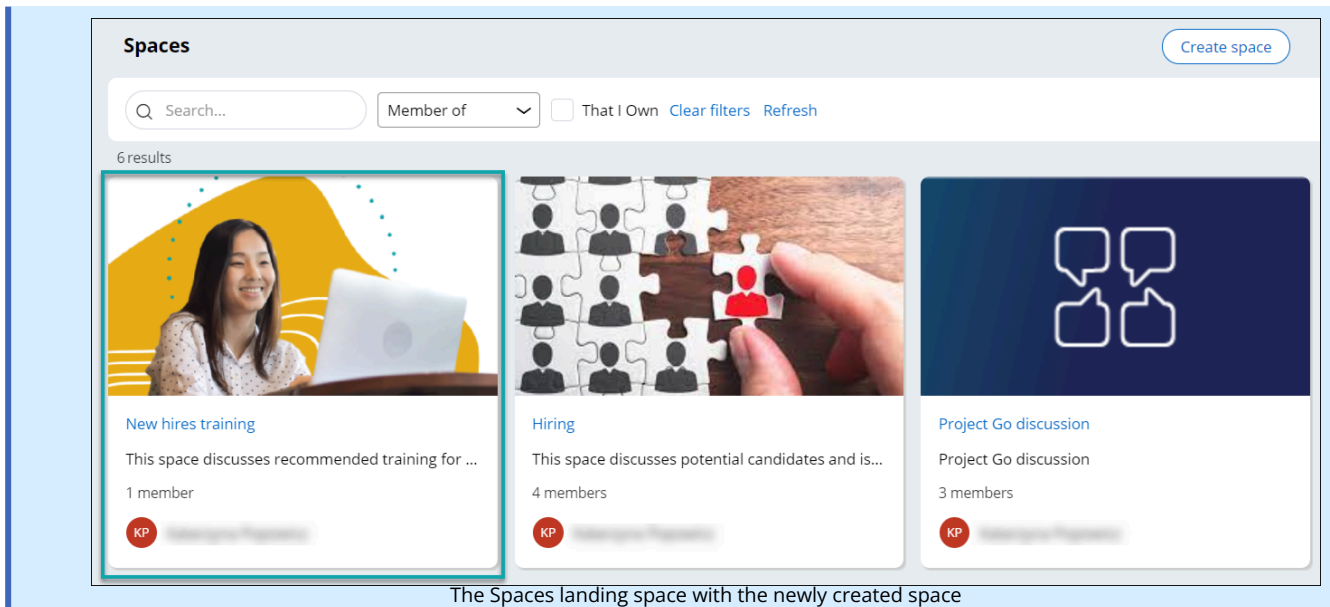
A space is created with a Pulse interface for discussions and sections that contain the space details, as shown in the following

image:



A newly created space interface

The newly created space is visible on the Spaces landing page, as shown in the following image:



## Joining a space

Join a space to discuss a topic of specific interest with other members, for example, to exchange views about prospective candidates for your team.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Join a selected space by doing one of the following actions:
  - To join a public space, in the space preview, click **Join**.

You can now instantly collaborate with members of the space.

- To join a private space, in the space preview, click **Request access**.

When the owner approves the request, you receive a notification, after which you can collaborate with members of the space.

- To join an unlisted space, wait for an invitation from the owner of the space.

When the owner invites you, you receive a notification, after which you can collaborate with members of the space.

👁 **Tip:** To leave a space, click **Actions > Leave**.

## Communicating with members of a space

Communicate with the members of a space by using Pulse to discuss relevant topics, for example, training for new hires.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space in which you want to start or continue a discussion.
4. On the **Activity** tab, use Pulse to communicate with members of the space.

For more information, see [Posting a message in Pulse \(on page 499\)](#), [Replying to a message in Pulse \(on page 503\)](#), and [Searching for Pulse messages in spaces \(on page 527\)](#).



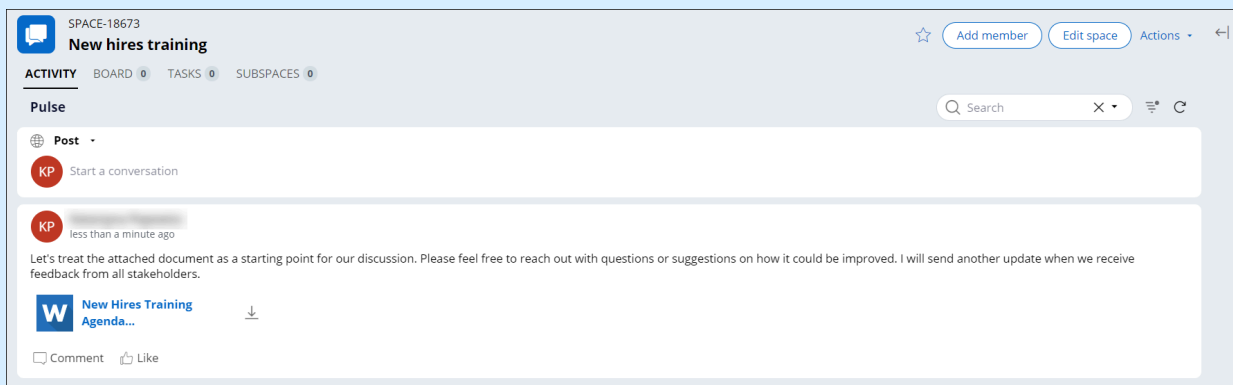
**Tip:** Select the types of notifications that you receive for the space by clicking **Actions > Notification preferences** and selecting the **Override at instance level** checkbox.

5. **Optional:** To add content to support your discussions, upload or pin files, URLs, documents, or cases and other spaces in your application.

For more information, see [Adding content to a space \(on page 519\)](#) and [Pinning content to a space \(on page 516\)](#).

6. If you are the owner of the space, moderate discussions and content by deleting messages, attachments, and pins that are irrelevant or inappropriate.

For more information, see [Managing Pulse messages \(on page 507\)](#).



Pulse message with an attachment posted in a space

## Pinning content to a space

Communicate with members of a space more effectively by pinning content that supports discussion in that space. For example, you can pin the job profiles of prospective candidates to a Hiring space. A pin is a reference to a file, URL, document, case, or space. You can add and edit titles and images for pins, search for pins, filter pins by pin type, and 'like' pins. You can also pin existing content in one space to another space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space to which you want to pin content.
4. On the **Board** tab, click **Add pin**.
5. In the **Create pin** modal dialog box, select a pin type.
6. Select the type of content to pin to the space:
  - For the **Case**, **Document**, or **Space** pin types, press the Down arrow key, and then choose a value from the field that appears, or upload a file.
  - For the **URL** pin type, enter a URL in the field that appears.
7. **Optional:** Update the pin title and image.

The title and image appear in the pin preview on the **Board** tab.

8. Click **Pin to space**.
9. **Optional:** To pin the current space to another space, click **Actions > Pin to space**, and then choose or create the spaces to which you want to pin the current space.
10. **Optional:** To pin the content to another space, perform the following actions:
  - a. Hover over the pin, and then click the **Show more** icon.
  - b. Click **Repin**.
  - c. In the modal dialog box, choose or create the spaces to which you want to pin the content.

The content that you pin appears on the **Board** tab of a space, which provides a comprehensive overview of the content that you link to the space.



**Tip:** You can reorganize pins to place the most relevant and the most important pins at the top of the list.

## Editing a pin in a space

Stimulate effective communication by updating the title and image of a pin, or adding an image to a pin to replace the default image. For example, you can add candidates' photos to the pins with job profiles in a Hiring space, to help the space members identify individual candidates.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space in which you want to edit a pin.
4. On the **Board** tab, search for the pin that you want to edit.
5. Hover over the pin, and then click the **Show more** icon.
6. Click **Edit**.
7. Update the pin title, image, or description.
8. Click **Submit**.

## Deleting a pin from a space

Delete a pin from a space when the pin is no longer relevant. For example, in a Hiring space, you can delete the job profiles of candidates who have not passed their interviews.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space in which you want to delete a pin.
4. On the **Board** tab, search for the pin that you want to delete.
5. Hover over the pin, and then click the **Show more** icon.
6. Click **Delete**.
7. Click **Submit**.

## Managing a space

Ensure that a space has correct information and relevant members by updating the details of the space and by adding and removing members. As an owner of a space, you can also approve or reject requests to join the space and set another member as the owner of the space.



The following tasks can help you manage a space:


## Updating details of a space

Update the details of a space to ensure that the information about the space is correct. For example, you can update the name of the space to be more specific and meaningful.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space for which you want to update the details.
4. Click **Edit space**.
5. In the **Edit space** modal dialog box, perform any of the following actions:
  - Update the name, description, type, or image for the space.
  - Enable or disable task tracking.
  - Update the hierarchy by adding, removing, or updating the parent space.

If you add a parent space, the current space becomes a subspace for the parent space. If you remove the parent space, the current space is no longer a subspace.

6. Click **Submit**.

 **Tip:** If the space is no longer required, delete the space by clicking **Actions > Delete**.

## Managing members of a space

Ensure that a space engages relevant stakeholders by adding and removing members. You can also approve or reject requests to join the space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space for which you want to manage the members.
4. Click **Add member**.

5. In the **Manage members** modal dialog box, perform any of the following actions:
  - To add a member to the space, in the drop-down list, press the Down arrow key to select the user that you want to add as a member, and then click the **Add member** icon.
  - To remove a member from the space, click the **Delete member** icon next to the name of the member.
  - To approve or reject member requests to join the space, click the **Approve member** or **Reject member** icon next to the name of the user.
6. Click **Submit**.

## Updating the owner of a space

Update the owner of a space if you do not want or need to own the space, or can no longer own the space. For example, if you are moving to another project in your organization, appoint a new owner of the space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space for which you want to update the owner.
4. On the **Activity** tab, click **Edit space**.
5. In the **Owner** section, click the **Change owner** icon next to the name of the current owner of the space.
6. Press the Down arrow key and select a member to set as the new owner.
7. Click **Submit**.

You no longer own the space, but you continue to be a member of the space.

## Adding content to a space

Attach relevant correspondence and documentation to a space so that you concentrate the supporting information in the space. For example, you can add sales orders from your customers so that the orders are available to other users of your space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.



3. Click the space to which you want to add content.
4. In the upper-right corner, click the **Expand the utility pane** icon, and then click the **Related** tab.
5. In the **Files & documents** section, click the **Manage files and documents** icon, and then in the **Manage content** modal dialog box, select the type of content that you want to add to the space:
  - Create a new document by using the rich text editor.
  - Add an existing document to reuse content from another space.
  - Add a URL, for example, the address of a customer's website.
  - Upload a local document.
  - Select a file from an external repository. For more information, see [Sourcing attachments from external storage \(on page 392\)](#).
6. Click **Submit**.

## Managing content attached to a space

Enhance collaboration with other users of your application by providing additional information about the attached documents in Pulse, and by performing other actions on the attached files that help you control the content. For example, you can prepare a draft of a presentation that your team is to deliver and discuss the content in Pulse.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Select a space that contains a document that you want to discuss.
4. In the upper-right corner, click the **Expand the utility pane** icon, and then click the **Related** tab.
5. In the **Files & documents** section, click the **Preview** link of the attached document.
6. Click the **Expand the utility pane** icon.
7. On the **Activity** tab, use the Pulse feed to discuss the document with other users.  
For more information, see [Posting a message in Pulse \(on page 499\)](#), [Replying to a message in Pulse \(on page 503\)](#), and [Managing Pulse messages \(on page 507\)](#).
8. In the document header, perform additional actions on the document by completing any of the following actions:

Choices	Actions
<b>Receive notifications for the document when you are not the owner</b>	Click the <b>Follow</b> icon.
<b>Modify the document</b>	Click <b>Edit</b> .



Choices	Actions
<b>Download the document</b>	Click <b>Download the file</b> .
<b>Add tags to the document</b>	Click <b>Actions</b> , and then select <b>Manage tags</b> .
<b>Modify your notification preferences for the document</b>	Click <b>Actions</b> , and then select <b>Manage notifications</b> .
<b>Pin the document to a space or to the Recents list</b>	Click <b>Actions</b> , and then select the relevant option.
<b>Copy a link to the document</b>	Click <b>Actions</b> , and then select <b>Copy shareable link</b> .
<b>Delete the comment</b>	Click <b>Actions</b> , and then select <b>Delete</b> .

## Fine-tuning collaboration by creating subspaces

Improve collaboration on a topic by branching the parent space. In this way you can create multiple subspaces that contain relevant subtopic content for their members, grouped under their parent topic. For example, you can divide the Loans space into subspaces for different loan types, such as Car Loans or Mortgage Loans. In these subspaces, case workers can access only the information for the loan types on which they are working.

You can add multiple subspaces to a parent space. However, a subspace can have only one parent space. You can also create subspaces within subspaces.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
  - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
  - To view only the spaces of which you are a member or an owner, click **Member of**.
  - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space within which you want to create a subspace.
4. Click **Actions > Add subspace**.
5. In the **Create a new subspace in Parent space name** modal dialog box, perform steps 4 (on page 512) through 9 (on page 513) in *Creating a space (on page 512)*.
6. **Optional:** To edit or adjust the subspace, perform any of the steps in *Managing a space (on page 517)*, *Adding content to a space (on page 519)*, and *Managing content attached to a space (on page 520)*.

You can find the newly added subspace on the **Subspaces** tab of the parent space and also on the Spaces landing page. The subspace is created with a Pulse interface for discussions, a link to the parent space above the title of the subspace, and sections that contain the subspace details.

## Creating and managing tasks in a space

Resolve tasks more effectively when collaborating with other users of your space by monitoring work on a task board. Add tasks to your board so that all the space members are well-informed about the assignments that they need to complete. You can notify all the members of the space by creating new tasks, updating the status of existing items, and adding checklists to tasks.

For example, a team of salespeople in a large company wants to prepare a presentation for their customers. Members of the team need a single tracking solution because they are located at different sites.

1. In the navigation pane of the portal that you use, click **Spaces**, and then choose the space that you want to update with tasks.
2. On the **Tasks** tab, in the **Open** column, click the **Create task** icon.
3. In the **Add task** modal dialog box, enter a name for the task.
4. Click **Create**.
5. **Optional:** To create more tasks, repeat steps 2 (on page 522) through 4 (on page 522).
6. **Optional:** To edit the task details, click the task, and then in the **Task details** modal dialog box, update the relevant items:

Choices	Actions
<b>Change the task name</b>	In the <b>Task name</b> field, provide a new name for the task.
<b>Change the task status</b>	In the <b>Status</b> field, select a new status for the task.
<b>Change the owner of the task</b>	In the <b>Assignee</b> field, select a new owner for the task.
<b>Change the due date of the task</b>	In the <b>Due date</b> field, select a new due date for the task.
<b>Provide additional details about the task</b>	In the <b>Additional details</b> field, enter relevant information about the task.
<b>Enable quick filtering and checking the progress of tasks by adding task categories</b>	a. In the <b>Category</b> section, in the <b>Add category</b> field, define the categories for the task by entering the name of the

Choices	Actions
	<p>category or selecting the category from the list.</p> <div data-bbox="938 369 1463 554" style="border: 1px solid #0070C0; background-color: #E6F2FF; padding: 10px;"> <p><b>Note:</b> The list only contains the categories that you create in the space that you currently use.</p> </div> <p>b. Click <b>Submit</b>.</p>
<p><b>Provide more detailed, smaller action items in the task</b></p>	<p>a. In the <b>Checklist</b> section, click the <b>Add checklist item</b> icon for every action item that you want to add.</p> <p>b. Click <b>Submit</b>.</p>
<p><b>Post a Pulse message about the task</b></p>	<p>In the Pulse section, in the <b>Start a conversation</b> field, enter your message. For more information, see <a href="#">Posting a message in Pulse (on page 499)</a>.</p>

7. **Optional:** To add content to a task, hover over the task, and then click **Show more > Add content:**

Choices	Actions
<p><b>Provide content in a rich text editor</b></p>	<p>a. In the <b>Manage content</b> modal dialog box, click <b>Rich text</b>.</p> <p>b. In the <b>Name</b> field, enter the name of the document, for example <code>Presenta-tion agenda</code>.</p> <p>c. In the <b>Content</b> field, create the document.</p> <p>d. Click <b>Submit</b>.</p>

Choices	Actions
<b>Upload a file</b>	<ol style="list-style-type: none"> <li>In the <b>Manage content</b> modal dialog box, click <b>Local file</b>, and then select a file that you want to upload.</li> <li>In the <b>Name</b> field, enter the name of the file, for example <code>Graphics - charts</code>.</li> <li><b>Optional:</b> To provide additional information about the file, in the <b>Description</b> field, enter relevant details.</li> <li>Click <b>Submit</b>.</li> </ol>
<b>Pin a document</b>	<ol style="list-style-type: none"> <li>In the <b>Manage content</b> modal dialog box, click <b>Add existing</b>.</li> <li>In the autocomplete field, press the Down arrow key, and then select a document from your application, for example <code>Sales results file</code>.</li> <li>Click <b>Add</b>.</li> <li>Click <b>Submit</b>.</li> </ol>
<b>Add a URL</b>	<ol style="list-style-type: none"> <li>In the <b>Manage content</b> modal dialog box, click <b>URL</b>.</li> <li>In the <b>Subject</b> field, enter the name that you want to use for the URL, for example, if you add a URL to a site of a prospective client, enter the client's company name.</li> <li>In the <b>URL</b> field, enter the Internet address, for example <code>http://uplustel-co.com</code>.</li> <li>Click <b>Submit</b>.</li> </ol>

8. **Optional:** To effectively process your case, on the **Tasks** tab, manage your tasks in one of the following ways:

Choices	Actions
<b>Change the priority of a task</b>	Drag the task up or down within a column.
<b>Change the status of a task</b>	Drag the task to one of the following columns:

Choices	Actions
	<ul style="list-style-type: none"> <li>• Open</li> <li>• In-Progress</li> <li>• On-Hold</li> <li>• Completed</li> </ul> <div style="border: 1px solid #00a0e3; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> You can also change the task status in the task properties, which you configure in the <b>Task details</b> modal dialog box of a task, in the <b>Status</b> field. You can also mark a task as completed by clicking the <b>Complete task</b> button.</p> </div>
<p><b>Filter tasks to display only tasks with a specific category</b></p>	<ol style="list-style-type: none"> <li>a. Click the <b>Filter</b> icon, and then select a category, for example <code>Collect data</code>.</li> <li>b. Click <b>Apply</b>.</li> </ol>
<p><b>Add content to a task</b></p>	<ol style="list-style-type: none"> <li>a. Hover over the task, and then click <b>Show more &gt; Add content</b>.</li> <li>b. Perform any number of actions from step 7 (on page 523).</li> </ol>
<p><b>Edit task details</b></p>	<ol style="list-style-type: none"> <li>a. Click the task, and then update relevant information, for example, change the assignee or due date of the task.</li> <li>b. Click <b>Submit</b>.</li> </ol>
<p><b>Delete a task</b></p>	<ol style="list-style-type: none"> <li>a. Hover over the task, and then click <b>Show more &gt; Delete</b>.</li> <li>b. Click <b>OK</b>.</li> </ol>
<p><b>Edit the checklist to inform other users about the progress of a task</b></p>	<ol style="list-style-type: none"> <li>a. Click the task, and then, in the <b>Checklist</b> section, select a checkbox next to an item that you want to mark as complete.</li> <li>b. Click <b>Submit</b>.</li> </ol>

Choices	Actions
<p><b>Post a Pulse message about the task</b></p>	<p>In the Pulse section, in the <b>Start a conversation</b> field, enter your message.</p> <p>For more information, see <a href="#">Posting a message in Pulse (on page 499)</a>.</p>

The task board in your space now contains tasks that you process and clear to complete the assignment. The tasks are enriched with attached content to facilitate communication between space members. You can also track and manage the progress of a case by updating the status of the tasks.

## Granting Super Admin privileges to users

Provide users with Super Admin privileges to manage and edit any space, even if they are not members of the space. By authorizing certain users to manage any space in your application, you can manage spaces more effectively and have greater control over the content that space owners and members create.

Users with Super Admin privileges can perform the following actions in any space:


- Mark a space as promoted.
- Edit a space: change the space type, edit the name or description, and change the owner and hierarchy.
- Delete a space.
- Create subspaces.

1. Create an access group in the application in which you want to grant Super Admin privileges to a user.

For more information, see [Creating an access group \(on page 499\)](#).

2. Create an access role for promoted spaces.

For more information, see [Creating an access role by using the access role dialog box \(on page 499\)](#).

 **Tip:** To save time, clone an existing access role.

3. Add the `PegaSocial-Group` class to the access roles.

4. Add the `pxSpaceAdmin` privilege with level 5 and give access level 5 to the following functions:

- Read
- Write
- Delete
- Run a report of rules and instances

Level 5 is the highest level.

5. Add the new access role to the new access group.

For more information, see [Adding a role to an access group](#) (on page [10](#)).

6. Add the new access group to the users to whom you want to give Super Admin privileges, and then make the newly created access group the default access group.

## Searching for Pulse messages in spaces

Access the information that you need more quickly by searching for Pulse messages within the context of a space. For greater accuracy, apply filters, such as keywords, and message authors. For example, you can filter messages that include the phrase `Loan request`.

Search results include private and public posts, comments, and attachments that match the filters that you apply. If the result is a comment, Pulse displays the complete thread so that you can see the context of the comment. For attachments, the filters evaluate both file names and contents.

1. In the navigation pane, click **Spaces**.
2. Open the space for which you want to search the Pulse messages.
3. On the **Activity** tab, in the **Search** field, click the **Expand search option** icon.
4. In the **Filter posts** dialog box, define the filters that you want to apply.
5. Click **Search**.

The results display posts, comments, and attachments that match the applied filters.

## Collaborating on shared content by using Documents

By creating documents in an application, you ensure that all users of the application have sufficient information to work in a business process and you reduce context switching by limiting the need to use external documents. You also save time by sharing the files across multiple cases and spaces.

For example, an approver for a Car Loan case can collaborate on an address verification document with other approvers. The approver can open the document from the Documents landing page or the case itself, and then use Pulse to consult other approvers.



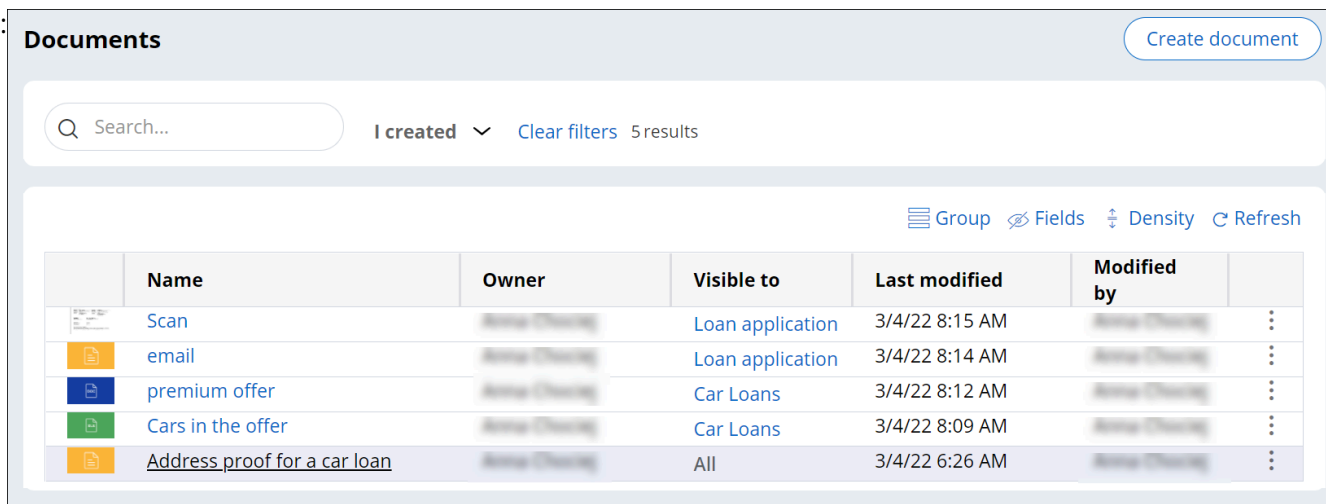
**Note:** The feature is not available in Cosmos React applications. You can only configure an application to launch the feature in a section-based UI in a new browser tab. For more information,

see Adding Theme Cosmos landing pages to the navigation menu (on page ) and Launching Theme Cosmos pages from Cosmos React portals (on page ).

Documents are files that contain additional information about cases and, unlike attachments, can be shared with all users in your application.

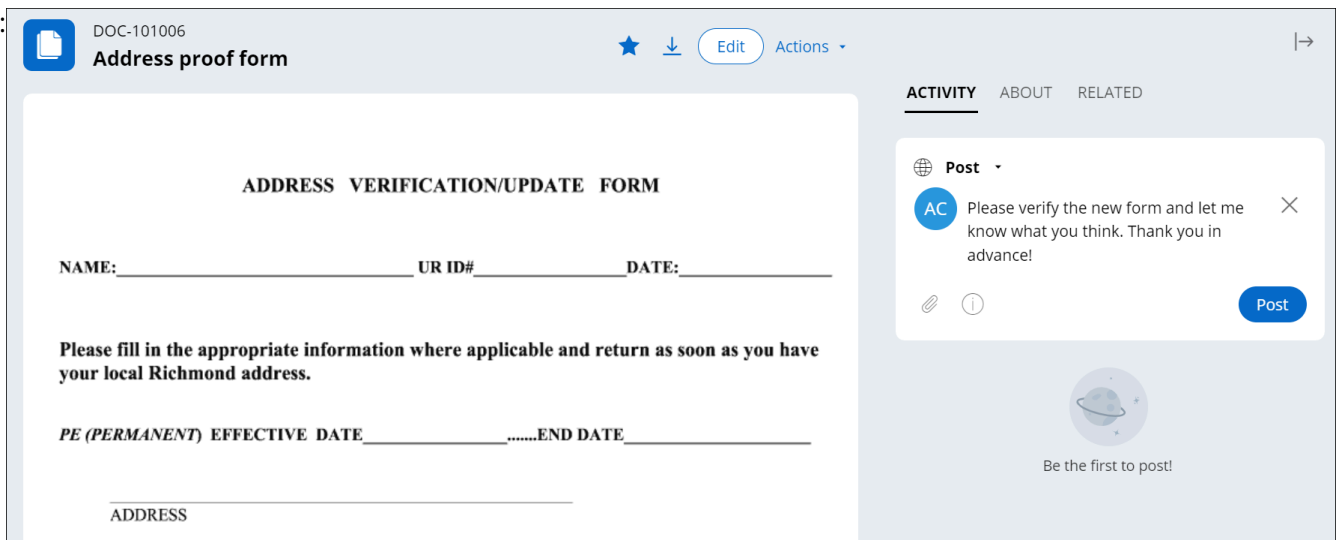
The Documents landing page in a portal consolidates all the important content for your application in one place.

The following image shows the Documents landing page in User Portal with all the documents that are available to a user within the application:



The Documents landing page in User Portal

The following image shows the collaboration interface for a document in User Portal:



The Pulse interface for a sample document



Documents are always created within the context of your application, and are not limited to a case, a space, or a Pulse conversation. However, you can reference the same document from multiple cases and spaces in your application.

You create documents directly in your application by using the built-in rich text editor or by uploading a file. When you create a document by uploading a file, you can provide a name for the document and use the rich text editor to add a meaningful description. If needed, you can also update documents later on.

With documents, you have the following options for collaborating on cases:

- Use Pulse to discuss a document with other users of your application.
- Follow documents.
- Pin documents to spaces.
- Link documents to cases.
- Add attachments to a document.
- Download a document.
- Generate a link to a document and share it with other users.

The following tasks can help you use documents to share information with other users:

## Creating a document in an application

Create a document to share or discuss information with other users in your application. For example, you can create a document and then use Pulse to discuss the document with your team.

The user who creates a document becomes the owner of the document.

1. Depending on the purpose of your document, create a document in one of the following places:

Choices	Actions
<p><b>To share general information, for example, the latest quarterly results of your organization, create a document on the Documents landing page.</b></p>	<ol style="list-style-type: none"> <li>a. In the navigation pane of your portal, click <b>Documents</b>.</li> <li>b. Click <b>Create document</b>.</li> <li>c. Enter a name for the document.</li> <li>d. On a selected tab, create document content in the rich text editor or select a file from a local or external repository or system. The options depend on your settings. By default, you can create a new docu-</li> </ol>

## Choices

## Actions

ment by using the rich text editor or upload locally saved documents.



**Note:** To enable users to download documents from external sources, configure external sourcing options in your application. For more information, see [Sourcing attachments from external storage \(on page 392\)](#).

## New Document



RICH TEXT



LOCAL FILE



CONTENT FROM

box

BOX.COM



REPO

Name \*

Job profile

To add information to a case, for example, an invoice in a Sales Order case, create a document from the case.

a. Open the case in which you want to create a document, and then click **Expand the utility pane**.

b. In the **Files & documents** section, click the **Manage files and documents** icon, and then in the **Manage content** dialog box, on a selected tab, create document content in the rich text editor, or select a file from a local or external repository or system.

The options depend on your settings. By default, you can create a new docu-

Choices	Actions
	<p>ment by using the rich text editor or by uploading a locally saved document.</p> <div data-bbox="938 369 1463 743" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p><b>Note:</b> To enable users to download documents from external sources, configure external sourcing options in your application. For more information, see <a href="#">Sourcing attachments from external storage (on page 392)</a>.</p> </div> <p>c. Click <b>Submit</b>.</p> <div data-bbox="938 856 1463 1066" style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p>When you upload a file or use the rich text editor, the system creates a document in the background.</p> </div>
<p><b>To add information to a space, for example, a job profile in the Hiring space, create a document from the space.</b></p>	<p>a. Open the space in which you want to create a document,</p> <p>b. On the <b>Activity</b> tab, click <b>Expand the utility pane</b> icon, and then click the <b>Related</b> tab.</p> <p>c. In the <b>Files &amp; documents</b> section, click the <b>Manage files and documents</b> icon, and then in the <b>Manage content</b> dialog box, on a selected tab, create document content in the rich text editor, or select a file from a local or external repository or system.</p> <p>The options depend on your settings. By default, you can create a new document by using the rich text editor or by uploading a locally saved document.</p>

Choices	Actions
	<p data-bbox="948 247 1458 575"> <b>Note:</b> To enable users to download documents from external sources, configure external sourcing options in your application. For more information, see <a href="#">Sourcing attachments from external storage (on page 392)</a>.         </p> <p data-bbox="906 659 1114 693">d. Click <b>Submit</b>.</p> <p data-bbox="1019 730 1445 898">           When you upload a file or use the rich text editor, the system creates a document in the background.         </p>

2. **Optional:** To provide only relevant content for the users, grant access to the document to members of a space or case by performing the following actions:
  - a. In the **New Document** dialog box, in the **Available to** section, select **Limited**.
  - b. In the **Select type** list, select **Space** or **Case**.
  - c. In the **Name** field, enter the name of a space or case.
3. Click **Publish**.

The document is added within the context of the application and is displayed on the Documents landing page.

## Discussing and managing a document

Discuss a document by using Pulse to review additional information about a case with other users in your application, for example, to analyze a faulty sales order.

**Note:** For applications that you build with the Theme Theme UI-Kit, see [Discussing a document](#).

1. In the navigation pane of your portal, click **Documents**.

The Documents landing page opens.

2. Click the document that you want to discuss.
3. On the **Activity** tab, use Pulse to discuss the document with other users.

For more information about using Pulse, see [Collaborating with users by using Pulse \(on page 499\)](#).

4. **Optional:** To modify the document by adding and managing additional elements, perform the following actions:

Choices	Actions
<p><b>To receive notifications for the document when you are not the owner</b></p>	<p>In the header of the document, click the <b>Star</b> icon.</p> <div data-bbox="862 804 1463 940" style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p><b>Note:</b> To unfollow the document, click the <b>Star</b> icon again.</p> </div>
<p><b>To modify the following elements:</b></p> <ul style="list-style-type: none"> <li>• <b>Name</b></li> <li>• <b>Attachments</b></li> <li>• <b>Description</b></li> <li>• <b>Visibility</b></li> </ul>	<ol style="list-style-type: none"> <li>a. In the header of the document, click <b>Edit</b>, and then modify any of the available options.</li> <li>b. Click <b>Publish</b>.</li> </ol>
<p><b>To save an offline copy of the document when the document is an uploaded file</b></p>	<p>In the header of the document, click <b>Download the file</b>.</p>
<p><b>To perform any of the following actions:</b></p> <ul style="list-style-type: none"> <li>• <b>View document history</b></li> <li>• <b>Add or manage tags</b></li> <li>• <b>Modify your notification preferences for the document</b></li> <li>• <b>Refresh the document page</b></li> <li>• <b>Copy a shareable link to the document</b></li> <li>• <b>Pin the document to a space or the recent items list</b></li> <li>• <b>Delete the document</b></li> </ul>	<p>In the header of the document, click <b>Actions</b>, and then select the relevant option.</p>

**Choices**

To add and manage the following items:

- Attachments
- Followers
- Stakeholders
- Tags

**Actions**

Click **Expand the utility pane**, and then, on the **Related** tab, edit the relevant section.

The screenshot shows a document interface for a document titled "Role description" (ID: DOC-103200). The document content area displays an attachment named "Role description.docx". A utility pane is expanded on the right side, showing the "RELATED" tab. This pane contains several sections: "Files & documents (1)" with a link to "Role description.docx"; "Followers (1)" with a user profile icon labeled "KP"; "Stakeholders (1)" with a user profile icon labeled "JS" and the name "John Smith" and status "Interested"; and "Tags (2)" with tags "#presentation" and "#team".

To add and manage references to cases, spaces, and other documents

Click **Expand the utility pane**, and then, on the **About** tab, edit the **Recent references** section.