

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.

- [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.

- [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.

- [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.

- [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.

- [Managing work across your team](#)

Achieve the best results while processing your Microjourney by efficiently managing work across your team. When you create a team for users and define routing conditions to assign tasks, you ensure that each task in the case is completed by the team member that is best-suited to the job. As a result, you meet your business goals faster, and to greater effect.

- [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.

- [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.

- [Enriching cases with external data](#)

Expand your Microjourney to take advantage of functionalities outside of your application, by connecting cases to external data sources. By enriching cases with external data, you can precisely cater to the needs of your customers, and resolve business cases more efficiently.

- [Designing surveys](#)

Gather information quickly from users while they process a case by designing a survey. When users complete a survey, the answers that they provide can capture the details that you need to successfully resolve your Microjourney.

- [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.

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.

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.

- [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.

- [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.

- [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.

- [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.

- [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.

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 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 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 about 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 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 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.

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](#).

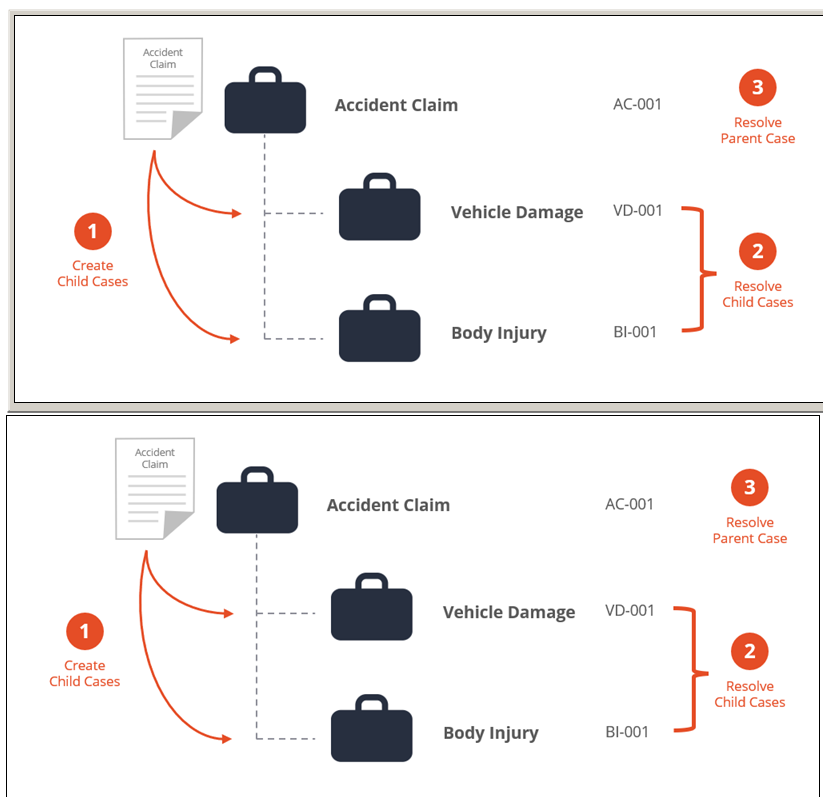
- [Creating a Microjourney for customer success](#)

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 a Car Accident 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.

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, filling in a complaint, and making a loan request. After a customer service representative (CSR) resolves each request, a customer receives a short survey to collect feedback. Collecting feedback is a separate process that includes creating a survey, 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. By reusing child case types in different scenarios, you speed up application development and provide 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](#).

- [Creating a case type](#)
- [Creating a top-level case type](#)
- [Creating a child case type](#)
- [Sharing data between parent and child cases](#)

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.

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

- [Defining case participants](#)
- [Notifying participants about events](#)
- [Mobile features overview](#)
- [Enabling case processing in offline-enabled mobile apps](#)
- [Conversational channels](#)

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:

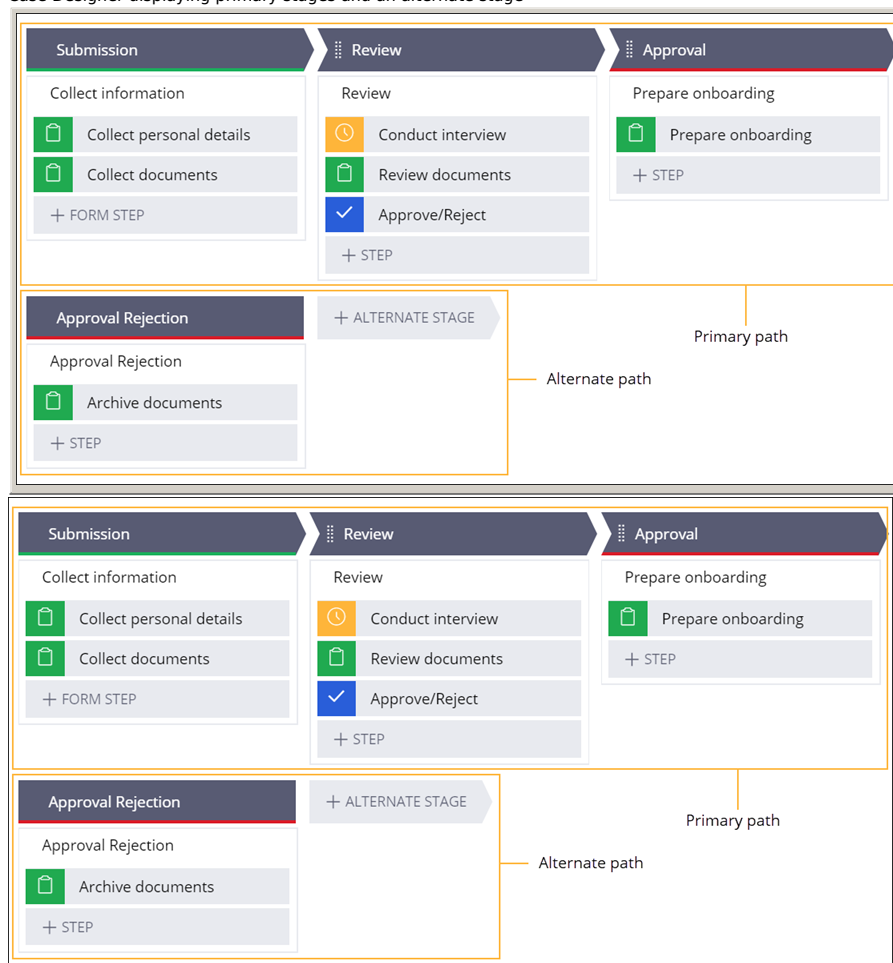
Stages

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.

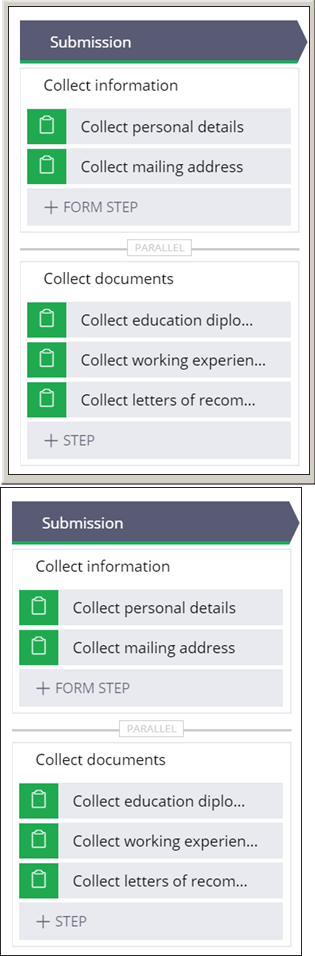
To define a stage, which represents a milestone in your business case, you create processes that are collections or individual tasks.

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.

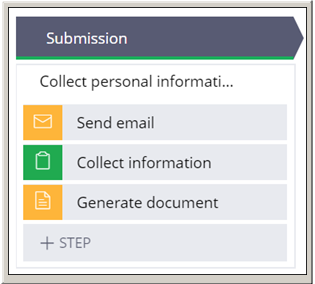
Steps

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 assignments 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](#).

- [Defining the case life cycle](#)
- [Adding a stage to a case life cycle](#)
- [Adding a process to a stage](#)
- [Adding a step to a process](#)

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.

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](#).

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 [Defining the case life cycle](#).

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 [Finding duplicate cases](#).

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](#) and [Validating field input](#).

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](#).

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.

For more information, see [Improving performance by archiving cases](#).

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](#).

- [Configuring case type settings](#)

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.

Before you begin: Review the business process that you want to automate, so that you can identify dependencies, determine logical paths for routing of work, and establish the required sequence of events.

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.

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.

- [Creating a case type](#)

Save time when processing similar business operations by defining case types that are the templates for work in your application. By creating case types, you automate business processes and support the different paths that case workers can follow to resolve a case.

- [Defining the case life cycle](#)

Track and resolve your cases in a convenient and facilitated way by defining the case life cycle that represents a business model. You can define the life cycle as smaller modules that match the elements of your business process.

- [Configuring a data model for a case](#)

Configure a data model to define data that you want to use in a case. For example, to include a user's address of residence in a case, add an address value to the data model.

- [Configuring views for cases](#)

Display relevant information and gather data from users by creating views for cases and associating the views with case types. 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.

- [Configuring case type settings](#)

Help your customers achieve success, and better respond to their individual needs, 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.

- [Configuring advanced processes](#)

Reach your business goals and solve cases that require additional processing by configuring advanced options for processes.

- [Modifying case types](#)

Modify your case types to reflect your unique business needs. For example, you can change a case type name, or delete a case type when you no longer need it.

- [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.

- [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.

- [Importing a case type](#)

To reduce the complexity of your application, share templates for business processes between your applications by importing case types.

Creating a case type

Save time when processing similar business operations by defining case types that are the templates for work in your application. By creating case types, you automate business processes and support the different paths that case workers can follow to resolve a case.

You can structure case types in a hierarchy to define parent-child relationships and share information among cases. To resolve a parent case, first resolve all of its child cases.

For example, you can define the following case types: Accident Claim, Vehicle Damage, and Bodily Injury.

A customer service representative (CSR) creates an Accident Claim case each time a customer reports an automobile accident. After your application verifies the customer's driving license and vehicle identification number, the system creates a Vehicle Damage case that is linked to the Accident Claim case.

The CSR decides whether the application needs a Bodily Injury case, based on information from the customer and the search results for similar cases. When other CSRs process the Vehicle Damage and Bodily Injury cases, the CSR enters funds payable and updated policy premium in the Accident Claim case.

A case manager then receives a notification that the Accident Claim is ready for review.

- [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.

- [Creating a child case type](#)

Create child case types to ensure that users of your application address all elements of 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.

- [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 a Microjourney for customer success](#)
- [Reclassifying a case type](#)
- [Deleting case types](#)
- [Renaming case types](#)

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 App Studio, click **Case types**.
2. In the header of the navigation pane, click **Manage**.
3. Click **New**.
4. In the **Case type name** field, enter a name for the case type.
5. **Optional:** To reuse assets from another case type, click **Advanced**, and then select a case type.
6. Click **Next**.

What to do next:

- Divide work into phases by adding stages to your case type. See [Creating a primary stage](#).
- Create child case types that represent dependencies that you resolve before the top-level case type can reach the resolution. See [Creating a child case type](#).
- [Creating a case type](#)
- [Defining the case life cycle](#)
- [Creating a Microjourney for customer success](#)

Creating a child case type

Create child case types to ensure that users of your application address all elements of 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 the review of an accident claim, and then supplement the process with child case types that visualize the estimating of vehicle damage and injuries. To resolve the Accident Claim case, a worker resolves the Vehicle Damage and Injury cases first. 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.
4. On the **Step** tab, 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 the case type.
5. **Optional:** To create more child case types, select **Create multiple cases using a list**, and then go to step 4.
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 source field that stores the information that you want to transfer.
 - b. In the **'To' field** column, in the text box that appears, enter the field in your child case where you want to use the information.
- d. **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**.

- [Creating a case type](#)
- [Reclassifying a case type](#)

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](#).

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. **For example:** Enter *Loan requests review*.
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.

For more information about your default work item, see [Setting your current 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 [Defining case participants](#).
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](#).
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](#).
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**.

- [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.

- [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.

- [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.

- [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.

- [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.

- [Introduction to case management](#)
- [Adding supporting processes to cases](#)
- [Finding duplicate cases](#)
- [Creating unit test suites with AUT](#)

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.

Before you begin: Create a parent case type by using a rule form. For more information, see [Creating 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. 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. **For example:** Enter *Run background check*.
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. **For example:** Enter *PhoneNumber*.
 - c. In the **To property value** column, enter a destination property in the child case type. **For example:** Enter *ContactNumber*. **Result:** At run time, your application populates the contact number in a *Run background check* child case type using a value that a user provided for a phone number in a *Review job candidate* parent case type.
 - d. **Optional:** To propagate values for more properties, repeat steps [5.a](#) through [5.c](#).
 - 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
Allow users to start a child case manually at run time	a. Select the Manual instantiation check box. b. Optional: To allow users to start the case only under certain circumstances, in the Permitted When field, enter a When rule that defines when a user can start a child case. For example: Select a When rule that allow users to start a <i>Run a background check</i> child case type only if a job candidate applies for a manager position.
Start a child case automatically	a. Select the Automatic instantiation check box. b. Select whether a child case starts together with a parent case or if a child case starts when a case fulfills the dependencies. c. Optional: If you configure a case type to start during a parent case creation, to allow case instantiation only under certain circumstances, in the Permitted When field, enter a When rule that defines when the child case starts. For example: Select a When rule that starts a <i>Run a background check</i> child case type only when a job candidate has less than five years of work experience.

Tip: You can select both manual and automatic start of a case type. If the system cannot 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. 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](#) through [7](#).
9. Click **Save**.

- [Understanding case hierarchy](#)
- [Creating a child case type](#)

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.
For example: Enter TotalInsuranceAmount.
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. **For example:** Enter VehicleDamageAmount.
8. **Optional:** To add more properties, click **Add property**, and then repeat steps 6 through 7. **For example:** Add a BodilyInjuryAmount property.
9. **Optional:** To add more aggregated properties, click **Add calculation**, and then repeat steps 5 through 7.
10. Click **Save**.

For example: The following figure shows a total insurance amount calculated by aggregating vehicle damage and bodily injury amounts:

Calculating aggregated properties

Aggregate properties

Property to calculate

Calculated property

TotalInsuranceAmount

Sum of following property values

Case type

Property

OJ0EAU-ReviewCI-Work-VehicleDamage

VehicleDamageAmount

OJ0EAU-ReviewCI-Work-BodilyInjury

BodilyInjuryAmount

+ Add property

Aggregate properties

Property to calculate

Calculated property

TotalInsuranceAmount

Sum of following property values

Case type

Property

OJ0EAU-ReviewCI-Work-VehicleDamage

VehicleDamageAmount

OJ0EAU-ReviewCI-Work-BodilyInjury

BodilyInjuryAmount

+ Add property

- [Validating case data](#)
- [Referencing properties](#)

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 [Adding a stage to a case life cycle](#).

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.

Choices	Actions
Move the case to the next stage	In the When all processes in this stage are complete section, select Automatically move to next stage . Note: This option is available for primary stages only.
Allow users to perform optional actions before moving the case to the next stage	In the When all processes in this stage are complete section, select Wait for a user action .
Resolve the case	a. In the When all processes in this stage are complete section, select Resolve the case . b. In the Resolution status field, enter a status that a case enters upon resolution. For example: To display a successful resolution status, enter Resolved-Completed. c. Optional: To delete any work that remains after the case resolution, select the Delete open assignments check box. d. Optional: To delete any dependent cases after the case reaches resolution, select the Delete open assignments check box.

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. **For example:** To display a status for an insurance case that enters a stage of reviewing gathered insurance data, enter Pending-Review.

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.
- For example:** You can validate format of personal details that users provide or require a document with an expertise of damage of insured goods.
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 through 9.e.

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

For example: The following figure shows a sample configuration for a process that starts after a When rule evaluates to true and that users need to resolve within a goal and deadline defined by a service-level agreement rule:

Configuring a process

The figure shows two screenshots of the 'Automatically launched processes' configuration window. The top screenshot shows a process named 'ReviewDamagePhotos' with 'When Rule' as the start process and 'Start upon stage entry' as the start step. The bottom screenshot shows the same configuration but with 'DamagePhotosAttached' as the start process.

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 through 10.c.
- For example:** You can add an optional process to review an insurance claim by a manager if an insurance amount exceeds a specific value.
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 through 11.c.
- For example:** You can add an optional action responsible for updating the personal details of a customer. If necessary, a customer service representative (CSR) can update the details without interrupting the main case processing.
12. Click **Save**.

- [Building case types](#)
- [Creating a primary stage](#)
- [Creating an alternate stage](#)
- [Adding a process to a stage](#)
- [Configuring and working with optional actions in case types](#)
- [When Condition rules](#)

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](#).

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.

Result: 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. To add more attachment categories, repeat steps 3 through 7.
9. Click **Save**.

For example: The following figure shows an attachment category for photos that are required in a case type to review vehicle damage insurance claims:

Attachment category configuration

Add attachment categories which are associated with Vehicle damage

Category	Description	Required for resolution	Property reference ?	
VehicleDamagePhotos	Visual materials for vehicle damage claim	☒	VehicleDamagePhotosAttachments	

+ Add attachment category

Add attachment categories which are associated with Vehicle damage

Category	Description	Required for resolution	Property reference ?	
VehicleDamagePhotos	Visual materials for vehicle damage claim	☒	VehicleDamagePhotosAttachments	

+ Add attachment category

- [Securing case access](#)
- [Controlling access to case attachments](#)
- [Referencing properties](#)

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

Before you begin: Create specifications that you want to associate with your case type. For more information, see [Creating a specification](#). 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. For more information, see [Tracking feature-driven development](#).

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 through 5.
7. Click **Save**.

- [Engaging with stakeholders to define a Microjourney](#)
- [Creating features](#)

Defining the case life cycle

Track and resolve your cases in a convenient and facilitated way by defining the case life cycle that represents a business model. You can define the life cycle as smaller modules that match the elements of your business process.

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

Define the case life cycle by completing the following actions:

- [Adding a stage to a case life cycle](#)

Visualize main elements of your business process by adding stages to a case life cycle. By creating stages, you organize work into sequential and logical phases that lead to achieving a goal of your business process.

- [Adding a process to a stage](#)

Organize tasks that are related in your business model by adding a process to a stage. With processes, you can control who performs work on a case, in addition to how and when the assignee resolves the case.

- [Adding a step to a process](#)

Represent a single assignment in your business process by adding a step to your case. 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.

- [Creating a Microjourney for customer success](#)
- [Creating a case type](#)

Adding a stage to a case life cycle

Visualize main elements of your business process by adding stages to a case life cycle. By creating stages, you organize work into sequential and logical phases that lead to achieving a goal of your business process.

For example, if your business goal is to review a job application from a prospective worker, you can divide the case into the following primary stages that represent the expected path of your case:

- Conduct screening
- Conduct interview
- Approve candidate

To ensure that you can resolve the process in an alternative way when an exception occurs, for example, if the candidate does not meet the requirements, create an alternate stage Approval Rejected.

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

- [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.

- [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 the cases that managers reject.

- [Defining the entrance criteria for a stage](#)

Evaluate case properties, attachments, and conditions to determine when a case enters a stage as it moves through the case life cycle.

- [Configuring 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.

- [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.

- [Removing a stage](#)

Remove a stage when it is no longer relevant to the life cycle of your case type.

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 Conduct screening, Conduct interview, and Approve candidate.

Before you begin: Define a case type. See [Creating a top-level case type](#).

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. If there are no primary stages defined, click **Add 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 panel, 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**.

What to do next: Organize work within stages by adding processes. See [Adding a process to a stage](#).

- [Defining the case life cycle](#)
- [Creating an alternate stage](#)
- [Adding a process to a stage](#)

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 the cases that managers reject.

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. Create an alternate stage: Click **Alternate stage**, and then enter a unique name in the text box.
 - To create a new alternate stage, in the **Case life cycle** section, click **Alternate stage**.
 - To convert a primary stage to an alternate stage, in the menu next to the stage name, click **Set as alternate stage**.
4. **Optional:** To let the case worker decide which stage the case enters after this stage, in the stage property panel, select **Wait for a user action**.
5. Click **Save**.

- [Defining the case life cycle](#)
- [Creating a primary stage](#)
- [Adding a process to a stage](#)

Defining the entrance criteria for a stage

Evaluate case properties, attachments, and conditions to determine when a case enters a stage as it moves through the case life cycle.

For example, you can check if an Auto 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:

- [Requiring attachments for stage entry](#)

Validate case attachments at run time to ensure that a case contains relevant files before it enters a stage.

- [Requiring property values for stage entry](#)

To ensure that a case contains relevant data before it enters a specific stage, validate the property values that users provide. For example, you can check if a job applicant provides contact details and working experience before a job application case enters the stage in which a manager approves or rejects the job application.

- [Creating a stage in the case life cycle](#)

Requiring attachments for stage entry

Validate case attachments at run time to ensure that a case contains relevant files before it enters a stage.

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 a stage name.
4. Click **Validation**.
5. Click + **Add required attachment** to display a list of attachment categories that are supported by your case type.
6. Select an attachment category from the list.
7. Click **Save**.

Result:

At run time, your application displays an error if the case does not have at least one attachment in each of the required attachment categories.

- [Creating a primary stage](#)
- [Controlling access to case attachments](#)
- [Creating a stage in the case life cycle](#)

Requiring property values for stage entry

To ensure that a case contains relevant data before it enters a specific stage, validate the property values that users provide. For example, you can check if a job applicant provides contact details and working experience before a job application case enters the stage in which a manager approves or rejects the job application.

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 **Validations**.
3. In the cell that corresponds to the row of the field to validate and the column of the stage for which you want to perform validation, click **Add entry validation**.
4. In the **Stage entry validation** window, in the **Message** field, enter the text to display when user input meets the validation conditions at run time.
5. Define the validation conditions for a field on the form:
 - a. **Optional:** To change the field that you want to validate, in the list of fields, select the name of the field.
By default, the system provides the field that you select in step 3.
 - 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.
The **Select values** option displays the fields and values from your form that you can use in the condition.

For example: To ensure that the birth date is not set in the future, select the birth date field from the list, set the **is after** comparator, and then select **Today** from the calendar control. When the user tries to submit the form with a birth date that is set in the future, an error message appears.

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

Choices	Actions
Ensure that the condition passes when all properties meet the criteria	a. On the right side of the row, click the Add a row icon. b. In the operator list, select and. c. In the list of fields, select the name of the field that you want to validate. d. In the comparator list, select the test that you want to perform on the field. e. In the value field, enter or select a value to compare against the user input. The Select values option lists the fields and values from your form that you can use in the condition.
Ensure that the condition passes when either property meets the criteria	a. On the right side of the row, click the Add a row icon. b. In the operator list, select or. c. In the list of fields, select the name of the field that you want to validate. d. In the comparator list, select the test that you want to perform on the field. e. In the value field, enter or select a value to compare against the user input. The Select values option lists the fields and values from your form that you can use in the condition.

For example: In a recruitment form, the user must indicate the time of employment at a specific company by entering dates in the **From** and **To** date type fields. To ensure that the start date is before the end date, and that the end date is after the start date, you can configure a validation condition with the AND logical operator.

7. **Optional:** To select additional stages for which you want to perform the validation, in the **Validate upon entering** list, select the stages.
By default, the system selects the check box for the stage that you select in step 3.
8. Click **Submit**.
9. **Optional:** To define more validation conditions, in the cell click **Add another validation**, and then repeat steps 4 through 8.
10. Click **Save**.

- [Defining the case life cycle](#)
- [Creating a primary stage](#)

Configuring 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.

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** if **Optional actions** is selected.
3. Decide which stage resolves the case, which is typically the last primary stage in the life cycle or an alternate stage.
4. Click the stage.
5. On the **General** tab of the property panel, click **Resolve the case**.
6. **Optional:** Change the default final status for resolved cases, by using one of the following techniques.
 - To use an existing status, press the Down Arrow key in the **Resolution status** field, and then select a status value.
 - To create a new status, enter a unique name in the **Resolution status** field by using the format *Resolved-[your status name]*.
7. **Optional:** Clean up dependencies when the current case is resolved.
 - a. Select the **Resolve open child cases** check box.
 - b. In the **Resolution status of child cases** field, enter a final status for child cases by repeating step 7.
8. Click **Save**.

Result: When all steps in the stage are completed, the case and its open child cases change status to the values that you provide. If you do not provide status values, the current case is set to *Resolved-Completed* and open child cases are set to *Resolved-Unspecified*.

- [Creating a stage in the case life cycle](#)

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.

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

Choice	Steps
Never skip a stage	a. In the Skip stage list, select Never.
Use a custom condition	a. In the Skip stage list, select Custom condition. b. Click the Configure conditions icon. c. In the Configure condition window, define a condition that starts the process. For example: Case status is equal to Pending-Blocked d. Optional: To add more conditions, click Add a row, and then repeat substep 4.c. e. Optional: To group the conditions, select operators in the list. f. Click Submit.
Use an existing condition	a. In the Skip stage list, select Existing condition. 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 last stage in the sequence.

5. Click **Save**.

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

- [Defining the case life cycle](#)
- [Creating a stage in the case life cycle](#)

Removing a stage

Remove a stage when it is no longer relevant to the life cycle of your case type.

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. Hover over the stage name.
4. Click the **Menu** icon and then click **Delete stage**.
5. Click **Save**.

- [Defining the case life cycle](#)

Adding a process to a stage

Organize tasks that are related in your business model by adding a process to a stage. With processes, you can control who performs work on a case, in addition to how and when the assignee resolves the case.

Create processes for your case life cycle by completing the following tasks:

- [Learning about case processes](#)

A process in a stage is a flow or a sequence of events that models a business process. By adding processes to a stage, you can define the paths that users take as they move a case through its life cycle.

- [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.

- [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.

- [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.

- [Defining conditions for starting a process](#)

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

- [Removing a process from a stage](#)

You can remove a process from a stage in your case type when it is no longer required.

- [Creating a Microjourney for customer success](#)
- [Removing a process from a stage](#)

Learning about case processes

A process in a stage is a flow or a sequence of events that models a business process. By adding processes to a stage, you can define the paths that users take as they move a case through its life cycle.

Each process has one or more steps. The type of steps and settings that you choose depend on the purpose of your process and on how the process relates to other processes in the stage.

For example, if your business case is to interview a job candidate, you can create an Interview process that you divide into the following tasks:

- Schedule interview
- Notify candidate
- Conduct interview
- [Adding a stage to a case life cycle](#)
- [Adding a step to a process](#)

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

Before you begin: Add stages to your case type. See [Creating a primary stage](#).

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, hover over a stage, and then click **More Add process New process**.
4. In a new text field, replace the default process label with a descriptive name.
5. **Optional:** To change the run-time order of a process, drag the process to a different position in the stage.
6. Click **Save**.

Result: At run time, the process starts when all steps in the previous process are completed. **What to do next:** Provide meaningful tasks for your process by adding steps. See [Adding single steps to processes](#).

- [Conditionally starting a process](#)

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.

For example, if your business goal is to review a job application, you can create a parallel process named Background check that you can conduct outside of the main process.

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, 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.
4. In the text field, replace the default process label with a descriptive name.
5. Click **Save**.
6. **Optional:** To prevent the process from running each time that a case reenters the stage, configure the settings:

Result: At run time, the process starts when a case enters the stage. **What to do next:** Add meaningful tasks to your parallel process by adding steps. For more information, see [Adding a step to a process](#).

- [Conditionally starting a process](#)

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.

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 multistep 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 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 **Step** section, 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 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](#).
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 [Adding a step to a process](#).
9. In the text field, replace the default step label with a descriptive name.
10. Click **Save**.

- [Conditionally starting a process](#)

Defining conditions for starting a process

Ensure that a process starts only when needed in a business case by defining conditions for running the process. By starting processes conditionally, you save time because 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.

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:

Choice	Actions
Always start a process	a. In the Start process (otherwise skip) list, select Always .
Custom condition	a. In the Start process (otherwise skip) list, select Custom condition . b. Next to the Start process (otherwise skip) list, click the Configure conditions icon. c. In the Configure condition window, select a condition, a comparator, and a value to compare with the condition. For example: Case status is equal to New d. Optional: To add more conditions, click Add a row , and then repeat the substep 3.c . e. Optional: To group the conditions, select comparators from the list. For example: To make only one condition compulsory, from the list, select OR . f. Click Submit .
Existing condition	a. In the Start process (otherwise skip) list, select Existing condition . 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**.

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

- [Changing the path of a process](#)

Removing a process from a stage

You can remove a process from a stage in your case type when it is no longer required.

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. Hover over a process name in a stage.
4. Click the **Delete** icon.
5. Click **Save**.

Result:

The process is disassociated from the stage but the underlying flow is not deleted. You can reuse the process later, by adding it to a different stage or process within a stage.

- [Calling one process from another process](#)

Adding a step to a process

Represent a single assignment in your business process by adding a step to your case. 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.

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

- [Learning about step types](#)
Each step in a process has a type, based on the functionality that it provides. By using a variety of step types, you can make your cases more interactive.
- [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.
- [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.

- [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.

- [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.

- [Requesting approval from users](#)

Enforce business policies in your workflows by adding an approval step to a process. In approval steps, users can approve or reject a case after reviewing the case details.

- [Adding a preconfigured step to a process](#)

You can add a step to a process that is preconfigured with functionality, such as sending an email or creating a case. By using preconfigured steps, you can design case types more efficiently.

- [Validating field input](#)

Ensure that the user input on a form meets the required conditions before the case moves to the next step in the life cycle. By validating field input, you check whether the data values that users enter are valid and whether the system can process the information correctly. For example, you can ensure that the customer provides their name and address before placing an order.

- [Creating a Microjourney for customer success](#)
- [Adding a process to a stage](#)

Learning about step types

Each step in a process has a type, based on the functionality that it provides. By using a variety of step types, you can make your cases more interactive.

A step is a unit of processing that is typically executed as part of an ordered sequence. Each step in a process has a type, based on the functionality that it provides. By using a variety of step types, you can make your cases more interactive.

For example, you can use steps in a Mortgage case 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 a step type when you define the a case life cycle.

The following step types are supported:

Collect information

Assigns a task, which you define in a form with fields, to a user or work queue.

Approve/Reject

Assigns a task to a user, typically a manager, to review case information and then decide whether the case is approved.

You can customize the form for this task.

Processes

Calls another process, or flow, from the current process.

User actions

Assigns a predefined task to a user or work queue, such as adjusting goal and deadline or changing stage.

Automations

Provides preconfigured functionality, such as sending an email or creating a case.

- [Defining the case life cycle](#)

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.

Before you begin: Add processes to stages in your case type. See [Adding a sequential process to a stage](#).

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 a process, click **Step**, and then select a step type that you want to add.
4. In the text field, replace the default step label with a descriptive name.
5. Click **Save**.

What to do next: Start planning your Microjourney by associating personas with your case type. See [Adding personas to organize users](#).

- [Adding a preconfigured step to a process](#)

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 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, hover over a process in a stage, and then click **Step Collect information**.
4. In the new text field, enter a unique name that describes the task.
5. In the properties panel, in the **Step** section, 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 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](#).

6. **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.
7. **Optional:** To describe the purpose of the task, in the **Instructions for user** field, enter a description.

8. Click **Save**.

What to do next: Define the fields on the form that users see when they perform the task. For more information, see [Building forms for case types](#).

- [Assignment shapes in processes](#)
- [Defining the case life cycle](#)

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.

- [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.

- [Sending automatic emails from cases](#)
- [Learning about approval by email](#)

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.

Before you begin: To give an external party a one-time directed web access to perform an action, ensure that the flow in which you want to use DWA already exists. For more information, see [Creating a top-level case type](#).

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.
For example: The following figure shows adding an access group for the external user:

The screenshot shows the 'Edit Operator ID: External User' form in the Pega Platform. The 'Profile' tab is active. The 'Contact Information' section includes fields for Title, First name, Last name, Full name (set to 'External User'), Position / job title, Phone, and Email (set to 'External'). The 'Application Access' section has a radio button to select a default access group. Under 'Access Group', 'PRPC:Unauthenticated' is selected, and 'ABCCorp:Administrators' is added and highlighted with a red box. The 'Application' column shows 'PegaRULES 8' and 'ABCCorp 01.01.01'. The 'Localization' section includes fields for Default locale and Time zone (set to 'America/New_York').

Note: For security reasons, give the external users only the minimum necessary access to the rulesets of your application.

- 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. **For example:** Enter `http://mysystem/prweb/`. **Note:** The value of the URL might reference a proxy server other than your Pega Platform server.
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 Accounts**.
 - b. Create an email account in the appropriate class group.

For more information, see [Creating an email account](#).

- 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 a rule](#).
 - b. In the correspondence rule form, in the **Label** field, enter a name that starts with **External**. **For example:** Enter `External_Corr`.
 - 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**. **For example:** The following figure shows adding the *ExternalInstructions* correspondence fragment:

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:

Edit Operator ID: External User

ID: External
RS: Pega-ProcessCommander
[Edit]

For more information about how to configure a case participant, see [Defining case participants](#).

- In the **Data transform** field, enter a name for a new data transform.
- 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.
- In the **Source** field, enter the operator ID for the external user. **For example:** The following figure shows a data transform for mapping a user identifier with the external user role:

- Click **Save**.
 - Select the **Create participant automatically when the case starts** check box.
 - Click **Done**.
 - Click **Save**.
- Configure the assignment shape:
 - In the navigation pane of Dev Studio, click **App**.
 - On the **Classes** tab, expand the case type in which you want to edit the assignment, and then click **Process Flow**.
 - In the **Flow** form, on the **Diagram** tab, double-click an assignment.
 - In the **Assignment properties** dialog box, in the **Routing** section, from the **Route to** list, select **Custom**.
 - In the **Assignment type** list, select **External**.
 - In the **Router** field, select **ToWorkbasket**, and then provide the name of the work queue to which you want to route the assignment.
 - In the **Advanced** section, click the **Assignment details** node, and then, in the **OperatorModel** field, select the operator ID for the external user.
 - 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.
 - In the **Party** field, enter the operator ID, for example `External`.
 - In the **Subject** field, enter the email message subject.
 - In the **HarnessPurpose** field, select **PerformExternal**.
 - Click **Submit**.
 - Click **Save**.

Result: 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.

- [Learning about operators](#)
- [Creating operator IDs](#)
- [Correspondence rules and email](#)
- [Configuring correspondence rules](#)
- [Flow shapes](#)

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.

Before you begin: Add a step to your case life cycle. For more information, see [Adding a step to a process](#).

- In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
- In the **Case life cycle** section, click the step that you want to supplement with instructions, for example, **Collect information**.
- In the **Step** property panel, click the **General** tab.
- In the **Instructions for user** field, enter additional information about the step. **For example:** Enter your personal details.
- Click **Save**.

Result: At run time, the instructions appear at the top of the form that the user completes.

- [Learning about step types](#)
- [Collecting information from a user](#)
- [Adding a process to a stage](#)

Requesting approval from users

Enforce business policies in your workflows by adding an approval step to a process. In approval steps, users can approve or reject a case after reviewing the case details.

For example, a manager can approve or reject a job candidate in a hiring process.

- In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
- On the **Workflow** tab, click **Life cycle**.
- In the **Case life cycle** section, in a stage where you want to add an approval step, click **Step Approve/Reject**. **Result:** Your case type includes an *Approval Rejection* alternate stage.
- In the text field that appears, enter a unique name that describes the step.
- On the **General** tab of the **Step** properties panel, 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, a user by user reference, the reporting manager of the user who last updated the case, or a case participant.

Note: You can route the approval step only to the participants that are of the application user type and represent a single user.

- 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.

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

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

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

Choices	Steps
Reject and move a case to the next step	a. In the If rejected then list, select Continue . b. In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.
Reject and change stage	a. In the If rejected then list, select Change stage . b. In the To list, select the stage that you want to assign to the case when a user rejects the case. c. In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.
Reject and resolve	a. In the If rejected then list, select Resolve . b. In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.

8. Click **Save**.

What to do next: To improve the turn-around time for your request, define a goal and deadline for the approval step. For more information, see [Setting service-level agreements \(SLAs\) for stages, processes, and steps](#).

- [Learning about approval by email](#)

Email approvals are a convenient way of requesting approval from stakeholders that use email messages to request and grant the approval. By configuring a case type to support email approval, your application's users can approve or reject case steps remotely, without having to log in to your application.

- [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.

- [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.

- [Enabling approval by email](#)
- [Setting service-level agreements \(SLAs\) for stages, processes, and steps](#)
- [Building forms for case types](#)
- [Defining the case life cycle](#)

Learning about approval by email

Email approvals are a convenient way of requesting approval from stakeholders that use email messages to request and grant the approval. By configuring a case type to support email approval, your application's users can approve or reject case steps remotely, without having to log in to your application.

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

- Case workers processing 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](#).

- [Enabling approval by email](#)

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

- [Requesting approval from users](#)

Enabling approval by email

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

Before you begin: Ensure that your case life cycle includes an approval step and that you have at least one email account in your application that can send and receive emails by performing one or more of the following tasks, as necessary:

- Add an approval step to your case life cycle. For more information, see [Requesting approval from a user](#).
- Create an email account. For more information, see [Creating an email account](#).

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 approval step for which you want to add email approval.
4. Select the **Enable email approval** check box.
5. Click **Save**.

- [Learning about approval by email](#)
- [Requesting approval from users](#)
- [Troubleshooting errors in unresolved cases](#)
- [Defining the case life cycle](#)

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.

Before you begin: Add an approval step to your case life cycle. For more information, see [Requesting approval from users](#).

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.

- In the **Step** properties panel, on the **General** tab, in the **Approval flow type** list, select **Cascading**.
- In the **Approval based on** list, select **Reporting structure**.
- 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**.
- Specify how many levels of approval a case requires:

Choices	Actions
One level of approval	Expand the Approval level section, and then select One .
Request approval from managers of all levels in the organizational chart	Expand the Approval level section, and then select All .
Create custom approval logic	- Expand the Approval level section, and then select Custom. - Click Update custom levels. - In the window that appears, click Add custom approval. - In the When field, enter a when condition. The when condition returns a Boolean expression that determines the manager required to approve the case. - In the Levels of approval field, enter how many levels of approval a case requires. Optional: To add more conditions, repeat steps 6.c through 6.e. - Click Submit.

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

Choices	Actions
Approve and move case to the next step	- In the If APPROVED then list, select Continue. Optional: To change the case status, in the Set status field, select or enter the status that you want to assign to the case when a user gives their approval.
Approve and change stage	- In the If APPROVED then list, select Change stage. - In the To list, select the stage that you want to assign to the case when a user gives their approval. Optional: To change the case status, in the Set status field, select or enter the status that you want to assign to the case when a user gives their approval.

- Define what happens when a user rejects the case:

Choices	Steps
Reject and move case to the next step	- In the If REJECTED then list, select Continue. - In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.
Reject and change stage	- In the If REJECTED then list, select Change stage. - In the To list, select the stage that you want to assign to the case when a user rejects the case. - In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.
Reject and resolve	- In the If REJECTED then list, select Resolve. - In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.

- Click **Save**.

What to do next: Provide more flexibility for users of your application and accelerate case resolution by enabling approval by email. For more information, see [Enabling approval by email](#).

- [Completing work on time](#)
- [Sending event notifications from cases](#)

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.

Before you begin:

- Add an approval step to your case life cycle. For more information, see [Requesting approval from users](#).
- Configure a page list property to hold the list of users who are required to give their approval, and a single-value property as an element of the page list to identify each user in the list. For more information, see [Properties](#).
- Optional:** Create a decision table that defines the users required to give their approval. For more information, see [Creating decision tables](#).

- In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
- In the **Case life cycle** section, click the approval step that you want to edit.
- In the **Step** properties panel, on the **General** tab, in the **Approval flow type** list, select **Cascading**.
- In the **Approval based on** list, select **Authority matrix**.
- Define the users required to give their approval in the step:
 - 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.
 - In the **Page list property** field, enter a page list that stores the users giving their approval.
 - In the **Approver property** field, enter a property that references the approving users.
- In the **Step** properties panel, click the **Flow** tab, and then define what happens when a user approves the case:

Choices	Actions
Approve and move case to the next step	- In the If APPROVED then list, select Continue. Optional: To change the case status, in the Set status field, select or enter the status that you want to assign to the case when a user gives their approval.
Approve and change stage	- In the If APPROVED then list, select Change stage. - In the To list, select the stage that you want to assign to the case when a user gives their approval. Optional: To change the case status, in the Set status field, select or enter the status that you want to assign to the case when a user gives their approval.

- Define what happens when a user rejects the case:

Choices	Steps
Reject and move case to the next step	- In the If REJECTED then list, select Continue. - In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.
Reject and change stage	- In the If REJECTED then list, select Change stage. - In the To list, select the stage that you want to assign to the case when a user rejects the case. - In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.
Reject and resolve	- In the If REJECTED then list, select Resolve. - In the Set status field, select or enter the status that you want to assign to the case when a user rejects the case.

- Click **Save**.

What to do next: Provide more flexibility for users of your application and accelerate case resolution by enabling approval by email. For more information, see [Enabling](#)

[approval by email.](#)

- [Completing work on time](#)
- [Sending event notifications from cases](#)

Adding a preconfigured step to a process

You can add a step to a process that is preconfigured with functionality, such as sending an email or creating a case. By using preconfigured steps, you can design case types more efficiently.

The following tasks can help you add a preconfigured step to a process:

- [Attaching content to a case](#)

You can use the Attach Content shape to control the kind of information that is added to a case. By attaching relevant correspondence and documentation, you can centralize the supporting information for a case.

- [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.

- [Creating a contextual case](#)

You can use the Create case shape to create a case in the context of your current case. By defining relationships, you can share data among cases in your application.

- [Collecting digital signatures in a case](#)

You can use the Send via DocuSign shape to collect digital signatures for your case attachments. By automating the review process for documents, you can save time and resolve cases more quickly.

- [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.

- [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.

- [Moving a case to a different stage](#)

You can use the Change to a specific 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.

- [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.

- [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.

- [Posting messages to a case](#)

You can use the Post to Pulse shape to add messages to the social activity stream of a case. By posting information as it becomes available, you can inform case workers and provide a visible discussion thread for stakeholders.

- [Preloading a data page](#)

You can use the Load data page shape to preload a data page that displays information to a user in a screen. By preloading a data page, you reduce the time that it takes for a screen to display information to a user.

- [Running a background process](#)

You can use the Run in background shape to start a background process, such as asynchronous processing or queuing items. The Run in background shape uses a Queue Processor rule for background processing.

- [Running a survey in a case](#)

You can use the Survey shape to collect information from a user who is processing a case. By providing a structured format for questions and answers, you can quickly incorporate user feedback into a case.

- [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.

- [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.

- [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.

- [Saving data in a data page as part of a flow](#)

You can save data that is in a data page to a system of record by including a Save Data Page smart shape in the flow of an application. By using this smart shape in a flow, you can save the data in an independent transaction, which enables the save data process to succeed or fail independently of the work item.

- [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.

- [Defining the case life cycle](#)
- [Creating a Microjourney for customer success](#)

Attaching content to a case

You can use the Attach Content shape to control the kind of information that is added to a case. By attaching relevant correspondence and documentation, you can centralize

the supporting information for a case.

1. Add the Attach Content 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, click **Life cycle**.
 - c. In a stage, click + **Step More Automations Attach content**.

Tip: For advanced processes, click **Configure process** to add the shape to the flow canvas.

- d. Click **Select**.
2. Attach content:
 - Attach a document by clicking **Attach a file**.
 - Attach a URL by clicking **Attach a link**.
 3. Click **Save**.

- [Adding a preconfigured step to a process](#)

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 check the values that users submit in the first stage of a case to ensure that the case reaches its resolution with the most appropriate effect. **Note:** PDF is a restrictive format that forces hard constraints on settings such as margins or layouts. Consequently, the *HTML to PDF* 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.

1. Add the Create PDF shape to the 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 a process in which you want to add the shape, click **Step More Automations Create PDF**.
 - c. 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. **For example:** To create a PDF file that contains case attachments, select **Case attachments**.
3. In the **PDF orientation** section, specify the file layout.
4. In the **PDF name** field, enter a descriptive name for the document. **For example:** Enter *Attachments for a job application process*.
5. In the **Attachment category** list, select an option that describes the business classification of the screen capture. **For example:** To indicate that the file contains documents that a job applicant issues, select **Document**.
6. **Optional:** In the **Audit note** field, enter the audit note.
7. Click **Save**.

Result: 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.

- [Adding a preconfigured step to a process](#)

Creating a contextual case

You can use the Create case shape to create a case in the context of your current case. By defining relationships, you can share data among cases in your application.

1. Add the Create Case 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, click **Life cycle**.
 - c. Hover over a process in a stage, and then click + **Step More Automations Create Case**.
 - d. Click **Select**.
2. Choose an option from the **Create the following case** list.
 - [*Case type*] - creates an instance of the case type that you select.
 - **Create new Case type** - creates an instance of a new case type, based on the value that you provide in the **Case type name** field.

By default, instances of new case types are children of the current case.

3. To create more than one case, provide a list.
 - a. Select the **Create multiple cases using a list** check box.
 - b. In the **List field** field, press the Down Arrow key, and then select a field group, which is a property of type **Page List**.
4. Propagate data from the current case to the new case.
 - a. Select the **Transfer information to new case** check box.
 - b. In the **Transfer information** dialog box, select the check box next to a field to copy its value from the current case to the new case.
 - c. **Optional:** To change which field stores the value in the new case, enter a field name in the **To field** field.
 - d. Click **OK**.
5. Enter the audit note.
6. Click **Save**.

- [Creating a case type](#)
- [Adding a step to a process](#)

Collecting digital signatures in a case

You can use the Send via DocuSign shape to collect digital signatures for your case attachments. By automating the review process for documents, you can save time and resolve cases more quickly.

Note: This shape requires a configured DocuSign account.

For more information, see *Integrating DocuSign with the Pega 7 Platform* on the Pega Community.

1. Add the Send via DocuSign shape to the life cycle of your case.
 1. On the **Workflow** tab, click **Life cycle**.
 2. Hover over a process in a stage, and then click + **Step > More > Utilities > Send via DocuSign**.

Tip: For advanced processes, click **Configure process** to add the shape to the flow canvas.

3. Click **Select**.
2. To change the default name of the page that stores the signature request, enter a top-level page in the Envelope field.
3. Click **Save**.

At run time, your application sends all case attachments to the DocuSign service. Signed documents are then reattached to your case.

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.

1. Add a Search duplicate cases shape to the life cycle of your case type. For more information, see [Searching duplicate cases](#).

- Define the criteria that determine whether a case is a duplicate by performing the following actions:
 - Define the basic conditions that potential duplicates must match exactly. For more information, see [Defining basic conditions for case types](#).
 - Define a threshold that potential duplicates must match or exceed, and then specify the weighted conditions that support that threshold. For more information, see [Defining a threshold with weighted conditions for a case type](#).
- Optional:** To transform a temporary case into a permanent object in your database when your application does not identify any duplicates, add a Persist Case shape to the life cycle of your case type. For more information, see [Persisting temporary cases](#).

Result: At run time, 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.

- [Searching duplicate cases](#)

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

- [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.

- [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.

- [Adding a preconfigured step to a process](#)

Searching duplicate cases

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

- 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, on the **Workflow** tab, click **Life cycle**.
- In a process to which you want to add a Search duplicate cases shape, click **Step More Automations Search duplicate cases**.
- Click **Select**.

What to do next: Define conditions that potential duplicate cases must match:

- Define the basic conditions. For more information, see [Defining basic conditions for case types](#).
- Define a threshold with weighted conditions. For more information, see [Defining a threshold with weighted conditions for a case type](#).
- [Creating 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.

Before you begin: Add a Search duplicate cases shape to the life cycle of your case type. For more information, see [Searching duplicate cases](#).

- 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, on the **Workflow** tab, click **Life cycle**.
- In the **Case life cycle** section, click the **Search duplicate cases** step for which you want to define basic conditions.
- In the step property panel, click **Add basic condition**.
- 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.
For example: To compare your case with all cases that contain the same email address, enter `Email is same`.
Note: Properties that an exact condition references must be exposed as columns in the database. For more information, see [Planning for property optimization](#).
- Optional:** To define more basic conditions, in the dialog box, click **Add a row**, and then go to step 5.
- Optional:** To mark the case as a potential duplicate only when the case meets all basic conditions, in the field below **When**, select **and**.
- 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**.
- Click **OK**.
- Click **Save**.

Result: 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.

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.

What to do next: Define the weighted conditions for the case. For more information, see [Defining a threshold with weighted conditions for a case type](#).

- [Adding a preconfigured step to a process](#)

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.

Before you begin: Add a Search duplicate cases shape to the life cycle of your case type. For more information, see [Searching duplicate cases](#).

- 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, on the **Workflow** tab, click **Life cycle**.
- In the **Case life cycle** section, click the **Search duplicate cases** step for which you want to define the weighted conditions.
- In the step property panel, click **Add weighted condition**.
- 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.
For example: When you create three conditions with the weight of 20 each, the total weight is 60.
Note: Properties that a weighted condition references must be included as columns in a version of the `pyCaseMatchProcessingTemplate` report definition that applies to the class of your case type. For more information, see [Planning for property optimization](#).
- 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, to determine duplicates.
The application uses these values to compare existing cases against the current case to identify potential duplicates.
For example: To compare your case with all cases that contain the same email address, enter `Email is equal to user1@example.com`.
- Click **OK**.
- Optional:** To add more weighted conditions, repeat steps 4 through 7.
- 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.
- Click **Save**.

Result: At run time, the application evaluates the conditions that have different weights to determine which cases are less or more 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.

- [Adding a preconfigured step to a process](#)

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.

Before you begin:

- Ensure that your application has data to populate a document by capturing case information. For more information, see [Collecting information from a user](#).
- Ensure that you can include all the tags that you need by defining a data model for your case. For more information, see [Configuring a data model for 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 a process to which you want to add a **Generate document** shape, click **Step More Automations Generate document**.
3. In the **Step** properties panel, select a document to use as a template:

Choices	Actions
Create a new template	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. e. Repeat steps 3.c and 3.d for the tags that you need, and then save the document. f. In the Choose template document section, click Upload. g. In the Upload file window, click Choose file, and then navigate to the text template. h. Click Open. i. Click Done.
Use an existing template	a. In the Choose template document list, select a template that you want to use.

4. In the **Document name** field, enter a descriptive name for the output document. **For example:** Name your document *Job offer*.
5. In the **Category** list, select a category for your document.
6. Click **Save**.

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

- [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.
- [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.
- [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.
- [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.
- [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.
- [Creating project documents for stakeholders in Dev Studio](#)

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": ".PropertyName", "type": "P"	Design time: <pegaref config=("{name": ".DateTimeOfReport", "format": "MM/dd/yyyy HH:mm:ss a" setTZ: "true" doNotDisplayTZ: "false" format: "MM/dd/yyyy" type: "P" name: ".PropertyName"}" />
	Display date only	"format": "MM/dd/yyyy"	
	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 timezone	setTZ: "false"	
Date only		"name": ".PropertyName", "type": "P", "format": "MM/dd/yyyy"	Design time: <pegaref config=("{name": ".DateOfReport", "format": "MM/dd/yyyy" type: "P" name: ".PropertyName"}" />

Type	When to customize	Configuration parameters	Run time: 09/19/2019 Example
Time only		"name": ".PropertyName", "type": "P"	Design time: <pegaref config=("{name": ".TimeOfReport", "format": "
	Not 24h format	"format": "hh:mm a"	
	24h format	"format": "HH:mm"	
	Display timezone	setTZ: "true"	
	Do not display timezone	setTZ: "false"	
Integer, decimal, double		"name": ".PropertyName", "type": "P"	Design time: <pegaref config=("{name": ".TotalCost", "scale": "2", "type": "P", "doGroupBy": "true", "groupBySize": "3"
	Display decimal places	"scale": "Any number between 0 and 10"	
	Group digits	"doGroupBy": "true", "groupBySize": "Any number between 2 and 5"	
	Use symbol for currency	"isCurrency": "true", "currency": "Symbol_val"	
	Use code for currency	"isCurrency": "true", "currency": "_val (Code)"	
	Use symbol and code for currency	"isCurrency": "true", "currency": "Symbol_val (Code)"	
Boolean		"name": ".PropertyName", "type": "P", "tValue": "ValueForTrue", "fValue": "ValueForFalse"	Design time: <pegaref config=("{name": ".PropertyName", "type": "P", "tValue": "ValueForTrue", "fValue": "ValueForFalse"}") Run time: Yes
Attachment		"name": ".AttachmentFieldProp.pxResults(1).pyAttachmentLink", "type": "A"	Design time: <pegaref config=("{name": ".AttachmentFieldProp.pxResults(1).pyAttachmentLink", "type": "A", "showLabel": "true"}") Run time: The application displays image attachments, and embeds attachments. Otherwise, the application adds attachments as OLE embedding (OLE).
Text		"name": ".PropertyName", "type": "P"	Note: Users can insert images of any size when t
	Text area	"format": "TA"	
	Rich text editor	"format": "RTE"	
Properties inside page list		"name": ".PageList().PropertyName"	Design time: <pegaref config=("{mode": "pagelist", "name": ".PageList().PropertyName"}") Run time: Row 1: <i>Value1</i> , Row 2: <i>Value2</i>
Properties inside page		"name": ".Page.PropertyName"	Note: Document generation supports embedded properties. Design time: <pegaref config=("{name": ".Page.PropertyName"}") Run time: <i>PropertyValue</i>
Section		"name": "SectionName", "type": "S"	Note: Document generation does not support skin styling a might not properly render Pega autoger Design time: <pegaref config=("{name": "SectionName"}") Run time: <i>Section name</i>
Correspondence		"name": "CorrespondenceName", "type": "C"	Note: Document generation does not support skin styling a might not properly render Pega autoger Design time: <pegaref config=("{name": "CorrespondenceName"}") Run time: <i>Correspondence name</i>

- [Generating case documents](#)

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](#)

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.
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 inside tables.

- [Generating case documents](#)

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	Supported in Pega Platform 8.1	Supported in Pega Platform 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 as a report definition or data page	Page	Yes	Yes	Pega Platform supports only single-level page lists, for example in the following format:
	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	Supported in Pega Platform 8.1	Supported in Pega Platform 8.2 and later	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>sheetsToParse</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>sheetWiseData</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 true, the application deletes the file after parsing.
<i>outputPageName</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	Supported in Pega Platform 8.1	Supported in Pega Platform 8.2 and later	Description
Supported file format	.xlsx	Yes	Yes	N/A
Column-level generation		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.
Supported template 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.
	String	Yes	Yes	
	Numeric	Yes	Yes	

Feature	Boolean	Yes	Yes	Description
	Name Formula	Supported in Pega Platform 8.1 No	Supported in Pega Platform 8.2 and later	
Supported Excel cell types	Date	No	Yes	Sample cell value: {.pxResults().linkToSheet HYPERLINK} where <i>linkToSheet</i> is in the following format: <i>Label SHEET SheetName CellPosition</i>
	Hyperlink	No	Yes	
Supported clipboard data pages	Text	Yes	Yes	
	Decimal	Yes	Yes	
	Double	Yes	Yes	
	Integer	Yes	Yes	
	Date	Yes	Yes	
	TimeOfDay	Yes	Yes	
	DateTime	Yes	Yes	
Properties from any clipboard page	TrueFalse	Yes	Yes	Pega Platform supports only single- level page lists, for example in the following format:
	Page	Yes	Yes	
	PageList	Yes	Yes	
Styles	Font, cell, color	Yes	Yes	The application clones styles from a template and applies 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	Supported in Pega Platform versions from 8.1 to 8.3	Supported in Pega Platform 8.4 and later	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 <i>true</i> , the application downloads the generated file.
<i>DeleteFileAfterDownload</i>	Yes	Yes	If the value is <i>true</i> , the server deletes the generated file after the application downloads the file.
<i>evaluateFormulasLater</i>	No	Yes	If the value is <i>true</i> , 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).

- [Case documents tags](#)
- [Integrating a case with an external resource](#)

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>pxGenerateWordDocument</i> API	Comments
<i>{pega:reference propertyreference }</i>	Yes	• Data references can vary as Silverlight reads data from BLOBs. • No support for localization in <i>pxGenerateWordDocument</i>
<i>{pega:attachment attachfile="propertyName.insertMethod.extension.pzInsKey"}</i>	No	• Use the tag <i><pegaref config="{name":"AttachmentLink","type":"A"}></i> instead. • <i>AttachmentLink</i> is the property reference that provides the <i>pzInsKey</i> of Data-WorkAttach-File or Rule-File-Binary. • No support for attachments other than images inside tables, footers, and headers.
<i>{pega:insert corr=corrname}</i>	Partial	• No support for correspondence inside tables, headers and footers.
<i>{pega:insert corrinline=corrname}</i>	No	• Convert <i>{pega:insert corrinline=corrname}</i> to <i>{pega:insert corr=corrname}</i>. • Alternatively, to add correspondence inline within a document, use <i><pegaref config="{name":"CorrespondenceName","type":"C"}></i>. • No support for correspondence inside tables, headers and footers.

Tag in Silverlight	Support in the <i>pxGenerateWordDocument</i> API	Comments
<i>{pega:binary imagefile= AppName.imageFileName.FileType}</i>	No	- Parse images by using <i><pegaref config= {"name": ".AttachmentLink", "type": "A"}></i>. <i>AttachmentLink</i> is the property reference the provides the <i>pzInsKey</i> of Data-WorkAttach-File or Rule-File-Binary.
<i>{pega:binary binaryfile=TemplateClass.TemplateName}</i>	No	No support for reading the template and parsing the tags inside the template, and then adding the template back to the target file.
<i>{pega:binary imagefilelocalpath="LocalPath" }</i>	No	N/A
<i>{pega:binary imagefilesserverpath ="ServerPath" }</i>	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

You can use the Change to a specific 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.

- Add the Change to a Stage specific shape to the life cycle of your case.
 - In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
 - On the **Workflow** tab, click **Life cycle**.
 - Hover over a process in a stage, and then click **+ Step More Automations Change to a specific stage** or **Change to the next stage**.
 - Click **Select**.
- If you selected **Change to a specific stage**, in the **Stage** field, select the stage a case enters when the Change to a specific stage shape is processed.
- Optional:** To clean up unresolved work, select the **Close open assignments from current stage on change** check box.

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

- Enter the audit note.
- Click **Save**.
 - [Creating a stage in the case life cycle](#)
 - [Adding a step to a process](#)

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.

- Add a Wait shape to the life cycle of your case:
 - In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
 - In a stage in which you want to add a Wait shape, click **Step More Automations Wait**, and then click **Select**.
- 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	Actions
Wait until all child cases reach resolution	- In the Wait type list, select Case dependency. - Select the Wait for all child cases to be resolved check box.
Wait until one or all child cases reach resolution	- In the Wait type list, select Case dependency. - In the Wait for (Case type) lists, define how many of the selected case types need to reach resolution before the process resumes. - Click To be resolved. Note: A case reaches resolution when its case status starts with the word "Resolved". Optional: To consider statuses only after the case reaches the Wait shape, select the Consider status after wait begins check box.
Wait until one or all child cases reach a specific status	- In the Wait type list, select Case dependency. - In the Wait for (Case type) lists, define how many of the selected case types need to reach a certain status before the process resumes. - Click To reach status. - In the To reach status field, press the Down arrow key, and then select a status value. Optional: To consider statuses only after the case reaches the Wait shape, select the Consider status after wait begins check box.
Wait until a set amount of time passes	- In the Wait type list, select Timer. - Click Set date/time interval. - Enter integers in one or more of the fields that appear, based on the interval of time that you want the process to wait.
Wait until a specific date or time	- In the Wait type list, select Timer. - Click Reference date/time. - In the Date/Time field, select a property that stores the date and time when you want the process to resume. Ensure that you select a property that references the date and time in the future.

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

- [Design-time limitations in App Studio](#)
- [Defining the case life cycle](#)
- [Assignment shapes in processes](#)

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.

Before you begin: Create a temporary case. For more information, see [Creating temporary 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 App Studio, click **Case types**, and then click the case type that you want to open.
 - b. In a process in which you want to add the shape, click **Step More Automations Persist case**.
 - c. 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. Click **Save**.

- [Adding a preconfigured step to a process](#)

Posting messages to a case

You can use the Post to Pulse shape to add messages to the social activity stream of a case. By posting information as it becomes available, you can inform case workers and provide a visible discussion thread for stakeholders.

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. On the **Workflow** tab, click **Life cycle**.
 - c. Hover over a process in a stage, and then click **+ Step More Automations Post to Pulse**.

Tip: For advanced processes, click **Configure process** to add the shape to the flow canvas.

 - d. Click **Select**.
2. Identify the user who is responsible for the post.
 - To create a post for the user who is currently processing the case, in the **User posting** list, select **Current operator**.
 - To create a post for another user:
 1. In the **User posting** list, select **Other**.
 2. In the list that is displayed, select a user ID.
3. In the **Message** field, enter the text for the post.
4. Enter the audit note.
5. Click **Save**.

Result:

At run time, a post is added to Pulse that other users can view or respond to.

- [Defining the case life cycle](#)

Preloading a data page

You can use the Load data page shape to preload a data page that displays information to a user in a screen. By preloading a data page, you reduce the time that it takes for a screen to display information to a user.

1. Add the Load data page step before a step in which the data page displays information on a screen.
 - 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, click **Life cycle**.
 - c. Hover over a process in a stage, and then click **+ Step > More > Utilities > Load data page**.

Tip: For advanced processes, click **Configure process** to add the shape to the flow canvas.

 - d. Click **Select**.
2. In the **Data Page Name** field, press the Down Arrow key and select the data page that you want to preload.
3. **Optional:** Specify values for any parameters that are passed to the data page, for example, UI elements or non-UI rules.
4. Click **Save**.

Running a background process

You can use the Run in background shape to start a background process, such as asynchronous processing or queuing items. The Run in background shape uses a Queue Processor rule for background processing.

1. In the navigation panel 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. From the **Type of queue** list, choose a type of the queue processor that you want to use. For more information about when to use standard or dedicated queues, see [Queue Processor rules](#):
 - Choose **Standard**, and then provide an activity that the rule executes.
 - Choose **Dedicated**, and then provide the queue processor that you want to use.

To use the dedicated queue processor, you have to create the rule first. For more information, see [Creating a Queue Processor rule](#).
6. From the **Lock using** drop-down list, choose a type of lock that you want to use by selecting one of the following options:
 - To open a page by using a pzInskey property and lock it, select **Primary page**.
 - To open a page by using a key defined on a property, select **Key defined on property**.
 - To open a page without a lock from the database by using the pzInskey property, select **None**.
7. In the **Alternate access group** field, enter the name of an access group that you want to use to resolve the activity.

Provide an alternate access group if you want to resolve the activity in a different context from the context that was used to queue the item.
8. Click **Save**.

Running a survey in a case

You can use the Survey shape to collect information from a user who is processing a case. By providing a structured format for questions and answers, you can quickly incorporate user feedback into a case.

1. Add the Survey 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, click **Life cycle**.
 - c. Hover over a process in a stage, and then click **+ Step More Utilities Survey**.

Tip: For advanced processes, click the **Configure process** icon instead, to add the shape to the flow canvas.

- d. Click **Select**.
2. In the **Select survey** list, select the name of a survey.
3. In the **Route to** field, press the Down Arrow key and then select the user who is responsible for answering the questions in the survey.
4. Click **Save**.

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 App Studio, click **Case types**, and then click the case type that you want to open.
 - b. 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
Create a new notification	a. Select Create new. b. In the Recipients list, select a type of recipient, and then define the recipient. For example: Select Email address, and then provide the address. c. Optional: To add more recipients, click Add recipient, and then repeat step 2.b. d. In the Message/Email subject 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. For example: "The case "+ .pyID + " has been assigned to you". e. If you send email notifications, in the Email content section, click Compose, and then compose the email message. For more information about composing email messages, see Sending email notifications from cases.
Reuse existing notification	a. Select Use existing. b. In the Notification name list, select the notification that you want to send.

3. Click **Save**.

- [Adding a step to a process](#)
- [Setting notification preferences](#)
- [Creating unit test suites with AUT](#)

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.

1. Add the **Send Email** 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 **Case life cycle** section, in a stage, click **Step More Automations Send email**.
 - c. Click **Select**.
2. In the **Step** panel, 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](#).
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 names in the title to make it more dynamic and meaningful, for example, "The case "+ .pyID + " has been assigned to you".
6. In the **Message content** section, define the email content by clicking **Compose**, and then editing the message in the rich text editor:

Choices	Actions
Compose your own message	a. Enter and style the text by using the rich text editor toolbar. For example: You can create a numbered list or emphasize text with bold font.
Populate your email message with text from a template	a. In the Compose message window, click Choose template. b. Click a template that you want to use. c. Click Select. After the template text appears in your email message, you can edit this content.
Include the value of a field from your case in your email	a. In the rich text editor, position the text cursor in the place where you want to add a property. b. On the toolbar, click the Insert property icon. c. Select a property that you want to use. For example: You can reference the first and last name in a greeting, or embed a project milestone date within a sentence. Tip: To prevent word clusters in your message, include a space before and after your property reference.
Provide instant access to the case from your email	a. In the rich text editor, position the text cursor in the place where you want to add a link. b. On the toolbar, click the Link icon. c. Select the Link to current case check box. d. Enter the display text and title for the link. e. Click OK.

7. Click **Done**.
8. **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
Include all case attachments	a. Select All case attachments .
Include only specific case attachments	a. Select Choose attachments. 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. Optional: To add multiple attachments, click Add attachment, and then select another category or field.

9. Click **Save**.

Result: 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 a preconfigured step to a process](#)
- [Categorizing case attachments](#)
- [Setting notification preferences](#)

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 App Studio, click **Case types**, and then click the case type that you want to open.
 - b. In a stage in which you want to add a Push notification shape, click **Step More Automations Push notification**, and then click **Select**.
 - c. Click **Select**.
2. In the **Step** properties panel, identify the recipients of the notification:

Choices	Actions
Notify users of an application	a. In the Recipient type list, select Application . Result: The system autopopulates the To section with the name of your application.
Notify users in an access group	a. In the Recipient type list, select Role . b. In the To list, select an access group.
Notify a specific user	a. In the Recipient type list, select User . b. In the To list, select the identifier of the user that you want to notify.

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.

For example: Enter: "We have processed your loan application:" + .pyCaseID.
4. **Optional:** To give users an audible notification when they receive a push notification, select the **Sound alert** check box.
5. **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.
6. Enter an audit note.
7. Click **Save**.

- [Defining the case life cycle](#)
- [Adding a step to a process](#)
- [Engaging mobile users with push notifications](#)

Saving data in a data page as part of a flow

You can save data that is in a data page to a system of record by including a Save Data Page smart shape in the flow of an application. By using this smart shape in a flow, you can save the data in an independent transaction, which enables the save data process to succeed or fail independently of the work item.

1. Configure a savable data page.
2. In the **Diagram** tab of a flow, click **Smart shape menu icon Smart Shapes Save Data page**.
3. Right-click the **Save Data Page** smart shape and click **View Properties**.
4. In the **Automation details** section, in the **Data Page Name** field, select or enter the name of the savable data page that you configure in [Step 1](#).
The selected or entered name must be an autopopulated property that references a savable data type.
5. **Optional:** To specify an autopopulated property, complete the following steps:
 - a. Select the **Use associated property** check box.
 - b. In the **Autopopulate Property Name** field, select or enter the name of an autopopulated property.
6. Click **Submit**.
7. Drag the **Save Data Page** smart shape to the part of the flow in which you want the data in the data page to be saved.
8. Click **Save**.

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 App Studio, click **Case types**, and then click the case type that you want to open.
 - b. On the **Workflow** tab, click **Life cycle**.
 - c. In a stage, click **More Automations Update a case**.

Tip: For advanced processes, click **Configure process** to add the shape to the flow canvas.
2. Click **Select**.
3. Indicate how many cases to update.
 - To update all child cases, from list, select **All**.
 - To update a specific child case, select the case from the list.
3. Click the **Gear** icon to select target fields and values to apply from the current case.
4. Click **OK**.
5. Enter the audit note.
6. Click **Save**.

- [Adding a preconfigured step to a process](#)

Validating field input

Applicable to **Theme UI-Kit** applications

Ensure that the user input on a form meets the required conditions before the case moves to the next step in the life cycle. By validating field input, you check whether the data values that users enter are valid and whether the system can process the information correctly. For example, you can ensure that the customer provides their name and address before placing an order.

Note: You define validation conditions that check whether the user input on the form is incorrect. If the user input matches the validation parameters, an error message appears when the user tries to submit the form.

Before you begin: Create a case type, and then define a case life cycle that has stages and processes with assignments. The forms in your assignments must have fields that you want to validate. For more information, see [Automating work by creating 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. In the **Case life cycle** section, select the step to which you want to apply the validation condition.
3. In the **Step** pane on the right, on the **General** tab, click **Configure view**.
4. **Optional:** To ensure that the user enters a value in a field before submitting the form, set the field as **Required**.
 - a. In the view configuration window, on the **Fields** tab, locate the field that you want to set as **Required**.
 - b. In the **Options** list for that field, select **Required**.
When the user tries to submit the form without a value in a required field, an error message appears.

For example: To prevent the user from submitting a form without entering the date of birth, set the birth date field as **Required**.
5. In the view configuration window, on the **Validations** tab, click **Add condition**.
6. **Optional:** To display an error message when the validation fails, in the **Message** field, enter the text of the message.
7. Define the validation conditions for a field on the form:
 - a. In the list of fields, select the name of the field that you want to validate.
 - 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.

The **Select values** option lists the fields and values from your form that you can use in the condition.

For example: To ensure that the birth date is not set in the future, select the birth date field from the list, set the **is after** comparator, and then select **Today** from the calendar control. When the user tries to submit the form with a birth date that is set in the future, an error message appears.

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

Choices	Actions
Ensure that the condition passes when all properties meet the criteria	a. On the right side of the row, click the Add a row icon. b. In the operator list, select and. c. In the list of fields, select the name of the field that you want to validate. d. In the comparator list, select the test that you want to perform on the field. e. In the value field, enter or select a value to compare against the user input. The Select values option lists the fields and values from your form that you can use in the condition.
Ensure that the condition passes when either property meets the criteria	a. On the right side of the row, click the Add a row icon. b. In the operator list, select or. c. In the list of fields, select the name of the field that you want to validate. d. In the comparator list, select the test that you want to perform on the field. e. In the value field, enter or select a value to compare against the user input. The Select values option lists the fields and values from your form that you can use in the condition.

For example: In a recruitment form, the user must indicate the time of employment at a specific company by entering dates in the **From** and **To** date type fields. To ensure that the start date is before the end date, and that the end date is after the start date, you can configure a validation condition with the AND logical operator, as in the following figure:

Validation of the employment date range

The screenshot shows a validation configuration window. At the top, there's a 'Message' field with the text 'Incorrect date range'. Below this, there are two rows of validation conditions. The first row has a 'From' field, an 'is after' comparator, and a 'To' field. The second row has a 'To' field, an 'is before' comparator, and a 'From' field. The two rows are connected by an 'and' logical operator. Each field has a 'Select values' link next to it. The entire configuration is labeled with a '1' in a blue circle.

9. **Optional:** To define more validation conditions, repeat steps 6 through 9.
10. Click **Submit**.

Result: At run time, the application evaluates all validation conditions. If any condition is met, an error message appears, and the user cannot submit the form until they correct the invalid fields.

- [Forms and fields](#)
- [Restricting user input in a field](#)

Configuring a data model for a case

Configure a data model to define data that you want to use in a case. For example, to include a user's address of residence in a case, add an address value to the data model.

Before you begin: Create a case type. For more information, see [Creating a top-level case type](#).

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**.
3. Enter a value name.
4. From the list, select a type of the value.
5. Click **Save**.

- [Validating case data](#)

Validate the values of properties in a case to help users provide correct information when they create or save a case.

- [Creating a Microjourney for customer success](#)

Validating case data

Validate the values of properties in a case to help users provide correct information when they create or save 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. On the **Settings** tab, click **Validation**.
3. Click the **Configure** link that is displayed next to the action that you want to validate.
4. On the Create form, provide the name and version of an unlocked ruleset.

Do not change the default values in the **Label** or **Apply to** fields because they are required for rule resolution.

5. Click **Create and open** to create the validate rule.
6. Complete the fields on the Validate form to define the relevant properties and conditions to evaluate and which error messages to display.
7. Click **Save** and close the Validate form.

- [Creating unit test suites with AUT](#)

Configuring views for cases

Applicable to **Theme UI-Kit** applications

Display relevant information and gather data from users by creating views for cases and associating the views with case types. 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.

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, the dynamic view displays the fields in the contextual data model during different 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:

- Create
- Prompts a user to provide information about a case, before processing the first step in the case life cycle.
- 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 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.

- [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.

- [Building forms for case types](#)

You capture user input in a case by building forms. When you create, configure, and organize relevant fields in a form, you can clearly communicate to users the tasks that they need to perform to process an assignment in a case.

- [Skipping forms to collect information before case processing](#)

Configuring dynamic views

Applicable to **Theme UI-Kit** applications

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.

Before you begin: To save time and apply consistent values to a case type, define fields in a case data model that you can include in a dynamic view. For more information, see [Configuring a data model for 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, click the **Views** tab, and then click a dynamic view.
Dynamic views display the **Dynamic view** icon.
3. In the dynamic view window, click **Edit this view**.
4. In the dialog window, update the dynamic view:
 - To add a preconfigured field, in the properties panel, 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 panel, 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 panel, 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.
5. Click **Submit**.
6. Click **Save**.

Result: When users see a dynamic view in a case, the view contains the fields that you configure.

- [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.

- [Configuring views for cases](#)
- [Building forms for case types](#)

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.

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.
3. In the settings properties panel, click **General**.
4. In the **Behavior** section, select the **Skip 'Create' view when users create a new case** check box.
5. Click **Save**.

Result:

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.

- [Configuring views for cases](#)
- [Forms and fields](#)

Building forms for case types

Applicable to **Theme UI-Kit** applications

You capture user input in a case by building forms. When you create, configure, and organize relevant fields in a form, you can clearly communicate to users the tasks that they need to perform to process an assignment in a case.

Create personalized forms by including single-value fields, field groups, and field lists. You can use simple fields that capture single pieces of information, for instance a name, surname, or a case ID. For more complex information, such as attachments or data references, include fancy fields in your forms. To categorize similar fields into one data entity, create field groups and field lists. When you build forms for your case types, you can save time by reusing existing elements, from single fields to complete forms. For example, you can reuse a field group to collect user's personal details, mailing address, phone number and email address, instead of creating each field separately. You can also embed a form that contains shipping and payment details that a user reviews in the final stage of a purchase order.

To meet your specific business needs, you can customize the forms that users complete. This includes adding new fields to your forms, such as single-value or field group fields.

- [Adding single-value fields to forms](#)

Collect specific 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.

- [Adding field groups to forms](#)

Save time and create comprehensive forms for your case types by categorizing related fields in a single data structure. For example, you can capture the first name, last name, and address of a person in the Prospective Client form in a single field group, instead of several individual fields.

- [Adding data references to forms](#)

Enhance your case processing by providing users with a list of data records that they can choose from when they complete a form. By using data references, you integrate the data with the cases in your application.

- [Adding field group lists to forms](#)

To create comprehensive forms, you can categorize related values in a single data structure by adding a field group list to your form. For example, you can create a list of related fields to capture a job applicant's previous employers, and proposed employment start and end dates. Customer service representatives then complete the information when they advance the case in the hiring process.

- [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.

- [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.

- [Forms and fields](#)
- [Configuring a form at run time](#)

Adding single-value fields to forms

Applicable to **Theme UI-Kit** applications

Collect specific 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.

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 box, click **Add field**.
6. In the **Field** text box, enter a unique name for the field.
7. In the **Type** list, define a field type:
 - To create a field that contains scalar properties, select a type from the **Simple** section.
 - To create a field that contains complex and custom properties, for example attachments, select a type from the **Fancy** section.
8. If more configuration options are available, perform the following actions for a field:

Choices	Actions
Add a text paragraph to the form	a. In the Type list, select Text (paragraph) . b. Click the Configure paragraph icon. c. In the dialog window, in the Display as list, define whether to display the paragraph as plain or rich text. d. Click Submit .
Add a picklist to the form	a. In the Type list, select picklist. b. Click the configure options icon. c. In the dialog window, in the Display as list, define a display mode for the picklist. d. In the Picklist options, define choices for the picklist. e. You can create your own choices or source them from a data view. Click Submit .
Add an attachment field to the form	a. In the Type list, select Attachment . b. Click the Choose the attachment category icon. c. In dialog window, in the Attachment category list , select the category. d. Click Submit .
Add a user reference to the form	a. In the Type list, select User reference . b. Click the Configure user reference icon. c. In the dialog window, in the Select record using list, define whether users can search for a user ID by using a search box or a drop-down list. d. Click Submit .

9. 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)**.
10. Click **Submit**.

What to do next: After you add a new field, the field becomes part of a data model for your case type, so that you can reuse the field in other forms. For more information, see [Reusing fields on forms](#).

- [Building forms for case types](#)
- [Reusing fields on forms](#)
- [Restricting user input in a field](#)

Adding field groups to forms

Applicable to **Theme UI-Kit** applications

Save time and create comprehensive forms for your case types by categorizing related fields in a single data structure. For example, you can capture the first name, last name, and address of a person in the Prospective Client form in a single field group, instead of several individual fields.

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, add a field group to your form:
 - To add an existing field group, in the properties panel click **Fields**, click **Add to view** next to a field group that you want to add, and then go to step [10](#).
 - To create a new field group, click **Add field**, and then go to step [6](#).
 When you create new field groups, you also add them to a data model that your cases uses.
6. In the **Field** text box, enter a unique name for the field.
7. In the **Type** list, select **Field group**.
8. In the **Options** list, define the display mode for the field group:
 - To preserve settings of individual fields, select **Auto**.
 - To make the fields read-only, select **Read-only**.
 Fields can be **Read-only**, **Optional**, or **Required**.
9. Add fields to the field group:

Choices	Actions
Reuse fields from a data model	a. In the list that appears, select the data model that you want to use. b. In the Views section, select the view that you want to use. c. Optional: To add more fields, next to the View list, click Open , and then add the fields. For more information about adding fields, see Adding single-value fields to forms . Note: Changes that you make to this view, affect other forms that also use this view.
Create new fields	a. In the list that appears, enter the name of a field group. b. In the View section, define whether the view is a default view. You can reuse default views in any case type in your application. c. Next to the View list, click Open , and then add the fields to the view. For more information about adding fields, see Adding single-value fields to forms .

10. Click Submit	Choices	Actions
-------------------------	----------------	----------------

- [Building forms for case types](#)

Adding data references to forms

Applicable to **Theme UI-Kit** applications

Enhance your case processing by providing users with a list of data records that they can choose from when they complete a form. By using data references, you integrate the data with the cases in your application.

For example, you can reference the Vehicle data object on an accident claim form to display a list of car models.

Before you begin: Define a data object. For more information, see [Defining the fields in a data object](#).

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 **Add field**.
6. In the **Field** text box, enter a unique name for the field.
7. In the **Type** list, select **Data reference**.
8. In the **Options** list, define the display mode for the data reference:
 - To indicate that users can optionally complete the field, select **Optional**.
 - To indicate that users need to provide a value in the field, select **Required**.
 - To indicate that users can only view the field, select **Read-only**.
9. In the list that appears, select a data object that you want to use.
10. In the **View** section, define a view for the data object:
 - To reuse a case view, select the name of the view.
 - To create a new view, select **Create new view**, click **Open**, and then add fields to the view.
 - To create a new default view, select **Create default view**, click **Open**, and then add fields to the view.
11. For more information about adding fields, see [Adding single-value fields to forms](#).
11. Configure a method for selecting prompt options by clicking the **Change the search fields** icon, and then performing one of the following actions:

Choices	Actions
Provide a search box	a. In the Select record using list, select Search box . b. In the Select in the following fields list, select a field that you want to include in the prompts. c. Optional: To include more fields in the prompts, click Add search field , and then repeat step 11.b . d. Click Submit .
Provide a list	a. In the Select record using list, select Drop-down list . b. In the Select using the following field list, select a field that you want to use. A value from this field populates the drop-down list. c. Click Submit .

12. Click **Submit**.

Result: After you add a data reference to a form, you can reuse the field because it is now a part of the data model for your case type. At runtime, your application displays the data reference as an autocomplete field, and then populates the field with a list of data records. Users can search for a data record by entering text in the field or selecting a record from a list.

- [Building forms for case types](#)
- [Referencing data from another data object](#)
- [Reusing fields on forms](#)
- [Referencing data from another data object](#)
- [Adding and configuring a Multi-select list control](#)

Adding field group lists to forms

Applicable to **Theme UI-Kit** applications

To create comprehensive forms, you can categorize related values in a single data structure by adding a field group list to your form. For example, you can create a list of related fields to capture a job applicant's previous employers, and proposed employment start and end dates. Customer service representatives then complete the information when they advance the case in the hiring process.

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 **Add field**.
6. In the **Field** text box, enter a name for the field.
7. In the **Type** list, select **Field group (list)**.
8. Define the fields in the field group list:

Choices	Actions
Reuse a field group list	a. In the last column, select a field group list that you want to reuse. b. To customize the list by adding more fields, click Add fields . For more information, see Adding single-value fields to forms .
Create a field group list	a. In the Field column of a new row, enter a name for the field. b. In the Type list, select a type for the field, for example Date only . c. In the Options field, select a display mode for the field. Fields can be Optional , Required , Read-only , and Calculated (read-only) . d. Optional: To add more fields, click Add field , and then repeat steps 8.a through 8.c .

9. Click **Submit**.

- [Building forms for case types](#)
- [Reusing fields on forms](#)

Reusing fields on forms

Applicable to **Theme UI-Kit** applications

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.

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](#).
8. Click **Submit**.

- [Adding field groups to forms](#)
- [Configuring dynamic views](#)
- [Reusing forms](#)

Reusing forms

Applicable to **Theme UI-Kit** applications

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.

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 **Views**.
6. Click **Add view** next to the view that you want to reuse.
7. **Optional:** To view or edit the form that you want to reuse, click **Open**.
8. 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**.

9. Click **Submit**.

- [Restrictions on field and form reuse](#)
- [Adding single-value fields to forms](#)
- [Supported display modes by field type](#)

Configuring case type settings

Help your customers achieve success, and better respond to their individual needs, 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.

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 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.

- [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.

- [Sharing data between parent and child cases](#)

Share data among cases to automate calculations and help case workers resolve cases more quickly.

- [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.

- [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.

- [Defining 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.

- [Notifying participants about events](#)

Escalate elapsed service-level agreements and highlight cases that require action by enabling email and push notifications for a case type. You can use push notifications for mobile devices, and set up email notifications for all devices that support email.

- [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.

- [Managing concurrent access to a case](#)

Set a locking strategy for parent case types to reduce the risk of losing updates to cases that users make simultaneously.

- [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.

- [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.

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.

- [Customizing the case icon](#)

Customize the icon for each case type in your application to help users identify cases more quickly.

- [Changing the case ID format](#)

Change the case ID format to help case workers distinguish between instances of different case types.

- [Geolocation tracking](#)

Geolocation tracking captures the longitude and latitude coordinates of case workers as they process a case. By default, this capability is enabled in the case types that you create.

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 App Studio, click **Case types**, and then click the case type that you want to open.
2. On the **Settings** tab, click **General**.
3. Click **Change icon**.
4. Click the icon that you want to select for your case type, and then click **Done**.
5. Click **Save**.

Result: The icon that you select overrides the previous icon provided on the Application Form.

- [Creating cases](#)
- [Changing the case ID format](#)

Changing the case ID format

Change the case ID format to help case workers distinguish between instances of different 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. On the **Settings** tab, click **General**.
3. In the **as Case ID prefix** field, enter a prefix for your case ID, for example, JobApp.
For more information on the ID format, see [Case ID format](#).
4. Click **Save**.

- [Creating cases](#)
- [Customizing the case icon](#)

Geolocation tracking

Geolocation tracking captures the longitude and latitude coordinates of case workers as they process a case. By default, this capability is enabled in the case types that you create.

Prerequisites

You can view geolocation information in a Google Map or case history. If you have access to Dev Studio, you can retrieve information from the *pyRecordHistoryList.pxResults* clipboard page.

Before using the geolocation tracking feature in your application, ensure that the following prerequisites are met:

- Consult local laws for possible restrictions on location tracking.
- Update the *uiengine/map/googleapikey* dynamic system setting with your Google Map API license key.
You must switch from App Studio to Dev Studio to update this setting.
- Accept your browser prompt, which is displayed when you process an assignment, to permit location tracking.
If you dismiss the prompt, it may disable geolocation tracking. You can refresh your browser to be prompted a second time.
- [Disabling geolocation tracking](#)
Disable geolocation tracking in a case type when the location of a case worker is not required in the audit trail of a case.
- [Disabling geolocation tracking](#)
- [Managing case history](#)

Disabling geolocation tracking

Disable geolocation tracking in a case type when the location of a case worker is not required in the audit trail 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. Click the **Settings** tab.
 3. Click **General**.
 4. Clear the **Enable geolocation tracking** check box.
 5. Click **Done**.
- [Managing case history](#)
 - [Geolocation tracking](#)

Sharing data between parent and child cases

Share data among cases to automate calculations and help case workers resolve cases more quickly.

Use the following techniques to share data between a parent and a child case:

- [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.
- [Data propagation](#)

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.

Before you begin: Create a child case type. For more information, see [Creating a child case type](#). 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**.

- In the **Propagate property value** field, press the Down arrow key, and then select the source property from which to copy data. **For example:** You can enter a full property name to specify an embedded property. For example, enter `.pyWorkParty(Customer).pyWorkPartyUri` to specify the `pyWorkPartyUri` property that is defined on the Customer page of the `pyWorkParty` page group.
- 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.
- Click **OK**.
- Optional:** To propagate the values of more properties from the parent case type, repeat steps 4 through 7.
- Optional:** To apply a data transform to set property values:
 - Select the **Apply data transform** check box, and then select a data transform to apply.
 - 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.
 - On the **Pages & Classes** tab, in the **Page Name** field, enter `pyParentPageName`.
 - In the **Class** field, enter a class that is equal to the class of the parent case.
 - On the **Parameters** tab, add a new entry with a name of `pyParentPageName`.
 - 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.
- Click **Save**.

Result: 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](#)
- [Creating a contextual case](#)

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:

- [Enabling automatic creation for child cases](#)

Enable automatic creation for a child case type to create a child case each time that a parent case is created.

- [Enabling conditional creation for child cases](#)

Enable conditional creation for a child case type to create a child case when a parent case is created and the case data meets the conditions that you define.

- [Enabling creation by email for top-level cases](#)

Associate an email account with a top-level case type to provide a way for users to create cases remotely.

- [Creating an Email channel](#)
- [Limitations on email discussions](#)

Enabling automatic creation for child cases

Enable automatic creation for a child case type to create a child case 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.

- 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 **Instantiation**.
- Select the **Automatically by system when** check box.
- Click **Parent case starts**. **Result:** 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.
- 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.
- Click **Save**.

Result: 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 true value.

- [Enabling conditional creation for child cases](#)
- [Enabling creation by email for top-level cases](#)
- [Configuring case creation](#)

Enabling conditional creation for child cases

Enable conditional creation for a child case type to create a child case when a parent case is created and the case data meets the conditions that you define.

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.

- 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 **Instantiation**.
Note: The **Instantiation** tab is available only for child case types.
- Select the **Automatically by system when** check box.
- Click **[Any or all] of the below conditions are met**.
- Click **Add condition** to define a dependency.
- In the **Case** field, press the Down Arrow key and select a case type on which your current case type depends.

This list is populated by case types that share your topmost case type and excludes: specialized case types, descendant case types, and case types with more than one parent.
- In the **Condition** list, select the criteria for creating a child case.
- If you select the **has work status** condition, select a case status in the list that is displayed.
Tip: When using work status as 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.
- If you select the **has completed** condition, click **any** or **all** to indicate how many cases must be resolved before a child case is created.
- Click **OK**.
- Optional:** Further refine the conditions that control whether a case is created by pressing the Down Arrow key in the **Allow only when** field and selecting the name of a when condition.
- Click **Save**.

Result:

At run time, a child case is created if one does not already exist, your dependency conditions are met, and the when condition that you provide returns a true value. The limitation of one child case prevents duplication when a dependency, such as a parent case type reaching a specific stage, is fulfilled more than once.

- [Changing case statuses](#)
- [Configuring case creation](#)

Enabling creation by email for top-level cases

Associate an email account with a top-level case type to provide a way for users to create cases remotely.

Before you begin:

- [Review the limitations on email accounts to understand how they affect your application](#)
- Do not use a personal email account because unread messages are deleted when the email listener is started.
- If you do not have an email account, contact your system administrator who can create one for you.
- Ensure that your system does not use LDAP authentication.

1. [Creating an email account in Dev Studio.](#)

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](#).
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](#).

- 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 **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.

- 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.

Result: When your application creates a case in response to an email message, the original message and corresponding replies are stored in Pulse.

• [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.

• [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.

• [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.

- [Disabling case creation by email](#)
- [Troubleshooting errors with email instantiation](#)
- [Example case discussion in email and Pulse](#)

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.

- [Example case discussion in email and Pulse](#)
- [Learning about approval by email](#)
- [Creating an email account in Dev Studio](#)

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.
4. 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](#). **Caution:** Do not delete the email account itself because another case type might rely on it.
5. In the case type rule form header, click **Save**, and then click **Actions Refresh**. **Result:** You disassociated the email account from your case type.
6. Create a new email account exclusively for your case type.
For more information, see [Creating an email account](#).
7. Enable email instantiation for your case type by using your new email account.
For more information, see [Enabling creation by email for top-level cases](#).

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

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. 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

4. 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](#).

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 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.

For example: The following figure shows a Java script in the *pyTreatIncomingText* activity before and after removing redundant tags from the email text:

Removing HTML tags from email messages

Label	Method	Step page	Description
1. Loop <input checked="" type="button" value="When"/>	Java		Removing the base tag from the message body, wh Jump Copy

Method Parameters

Java Source:

```

1 String messageString = myStepPage.getString(".pyInboundEmail.pyBody");
2
3 org.jsoup.nodes.Document doc = org.jsoup.Jsoup.parse(messageString);
4 doc.select("base").remove();
5 // remove xss related elements
6 /*
7 doc.select("script").remove();
8 doc.select("style").remove();
9 doc.select("form").remove();
10 doc.select("br").remove();
11 doc.select("meta").remove();
12 */
13 messageString = doc.body().text();
14
15 myStepPage.putString(".pyInboundEmail.pyBody", messageString);

```

Java Source:

```

1 String messageString = myStepPage.getString(".pyInboundEmail.pyBody");
2
3 org.jsoup.nodes.Document doc = org.jsoup.Jsoup.parse(messageString);
4 doc.select("base").remove();
5 doc.select("script").remove();
6 doc.select("style").remove();
7 doc.select("form").remove();
8 doc.select("br").remove();
9 doc.select("meta").remove();
10
11 messageString = doc.body().text();
12
13 myStepPage.putString(".pyInboundEmail.pyBody", messageString);

```

Label	Method	Step page	Description
1. Loop <input checked="" type="button" value="When"/>	Java		Removing the base tag from the message body, wh Jump Copy

Method Parameters

Java Source:

```

1 String messageString = myStepPage.getString(".pyInboundEmail.pyBody");
2
3 org.jsoup.nodes.Document doc = org.jsoup.Jsoup.parse(messageString);
4 doc.select("base").remove();
5 // remove xss related elements
6 /*
7 doc.select("script").remove();
8 doc.select("style").remove();
9 doc.select("form").remove();
10 doc.select("br").remove();
11 doc.select("meta").remove();
12 */
13 messageString = doc.body().text();
14
15 myStepPage.putString(".pyInboundEmail.pyBody", messageString);

```

Java Source:

```

1 String messageString = myStepPage.getString(".pyInboundEmail.pyBody");
2
3 org.jsoup.nodes.Document doc = org.jsoup.Jsoup.parse(messageString);
4 doc.select("base").remove();
5 doc.select("script").remove();
6 doc.select("style").remove();
7 doc.select("form").remove();
8 doc.select("br").remove();
9 doc.select("meta").remove();
10
11 messageString = doc.body().text();
12
13 myStepPage.putString(".pyInboundEmail.pyBody", messageString);

```

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. **For example:** The email provider might require a phone number to send you a verification code.

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](#)
- [Creating a Service Email rule](#)
- [Integrating with an email provider](#)
- [Creating an email listener](#)
- [Email account limitations](#)
- [Standard rules used by email instantiation](#)

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.

- [Email instantiation](#)
- [Troubleshooting errors with email instantiation](#)
- [Enabling creation by email for top-level cases](#)
- [Disabling case creation by email](#)

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.

For example: The following figure shows a sample set of values to use in your external application:

Sample code for creating a case by using an API

Generate create case microservice code

Call the following microservice in your external application to create a case of the current type.

Request URL (HTTP method POST) ⓘ
https://[redacted].com/prweb/api/v1/cases

HTTP authentication header ⓘ
Authorization: Basic [username:password Base64 encoded]

Request body ⓘ
{
 "caseTypeID": "DMOrg-DMSample-Work-ClearInteractionResults",
 "processID": "pyStartCase",
 "parentCaseID": "",
 "content": {}
}

Generate create case microservice code

Call the following microservice in your external application to create a case of the current type.

Request URL (HTTP method POST)

https://.com/prweb/api/v1/cases

HTTP authentication header

Authorization: Basic [username:password Base64 encoded]

Request body

{
"caseTypeID": "DMOrg-DMSample-Work-ClearInteractionResults",
"processID": "pyStartCase",
"parentCaseID": "",
"content": {}
}

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](#)
- [Enabling case processing in an external application](#)

Defining 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 panel, click **Participants**.
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. Define role preferences for the participant:

Choices	Actions
Participant with an account in your application	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. For example: If you map a participant to a current user, an application populates participant information with values that describe a user who currently processes the case at run time, without using a separate data transform for each user. c. Optional: 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 check box. d. Optional: To speed up case creation, enable automatic creation of participants when the case starts by selecting the Create participant automatically when the case starts check box.
Participant without an account in your application	a. In the Role has user account with this application section, select No. b. Optional: 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 check box.

8. Click **Done**.
9. Click **Save**.

- [Sending email notifications from cases](#)
- [Sending push notifications about case events](#)

Notifying participants about events

Escalate elapsed service-level agreements and highlight cases that require action by enabling email and push notifications for a case type. You can use push notifications for mobile devices, and set up email notifications for all devices that support email.

All assignments in the 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.
- A user transfers ownership of an assignment to another user.
- An application routes the assignment to the worklist of the user who processes the case.

- [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.

- [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.

- [Sending confirmations of completed assignments](#)

Inform users about completion of assignments by sending confirmation notes.

- [Sending custom notifications about new assignments](#)

To send custom correspondence to case participants about the creation of new assignments, override your notification policy.

- [Disabling assignment notifications](#)

Prioritize the tasks that need the greatest visibility in a case by disabling notifications for other assignments.

- [Creating unit test suites with AUT](#)
- [Creating an email account in Dev Studio](#)

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 App Studio, click **Case types**, and then click the case type that you want to open.
2. In the case work area, click the **Settings** tab.
3. In the settings panel, click **Notifications**.
4. In the **Email notifications** section, select the **Email user when assignment is routed to worklist** check box.
5. Define the content of your message:

Choice	Actions
Compose a new email	a. In the Email section, select Custom .
Use an existing email	a. In the Email section, select Use existing . b. In the Message list, select a message that you want to use.

6. In the **Subject** field, enter the email title by entering a string expression.
You can reference field names in the title to make the heading more dynamic and meaningful.
For example: "The case "+ .py/D + " has been assigned to you.".
7. If you create a custom message, in the **Message** section, click **Compose**, and then edit the message in the rich text editor:

Choices	Actions
Compose your own message	a. Enter and style the text by using the rich text editor toolbar. For example: You can create a numbered list or emphasize text with bold font.
Populate your email message with text from a template	a. In the Compose message window, click Choose template . b. Click a template that you want to use. c. Click Select . After the template text appears in your email message, you can edit this content.
Include the value of a field from your case in your email	a. In the rich text editor, position the text cursor in the place where you want to add a property. b. On the toolbar, click the Insert property icon. c. Select a property that you want to use. For example: You can reference the first and last name in a greeting, or embed a project milestone date within a sentence. Tip: To prevent word clusters in your message, include a space before and after your property reference.
Provide instant access to the case from your email	a. In the rich text editor, position the text cursor in the place where you want to add a link. b. On the toolbar, click the Link icon. c. Select the Link to current case check box. d. Enter the display text and title for the link. e. Click OK .

8. Click **Save**.

Result: A user receives an email when an application routes a task to their worklist.

- [Assigning tasks to users](#)
- [Assigning users automatically at run time](#)

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.

Before you begin:

- To use push notifications, install your mobile app on a device, and then enable push notifications in the app. For more information, see *Configuring a mobile app*, [Setting up push notifications in a mobile app](#), and [Configuring a mobile notification channel](#).
- To send push notifications about service-level agreements that mobile users miss, configure escalation actions for assignments. For more information, see [Defining an escalation action for a stage and a process](#).

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.
3. In the settings navigation panel, click **Notifications**.
4. 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. **Result:** 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.
5. Enable notifications about new assignments by selecting the **Notify user when assignment is routed to worklist** check box.
6. Click **Save**.

- [Assigning tasks to users](#)
- [Assigning users automatically at run time](#)
- [Completing work on time](#)

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.

For example: You can configure the *Create a purchase order* assignment with the following confirmation note: A purchase order will now be created.

6. Click **Submit**.
7. Click **Save**.

Result:

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](#)

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**.

Result: 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](#)

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](#)

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.

Before you begin:

- To see field-level changes at run time, ensure that your application uses the latest version of the UI Kit.
- Mark as relevant records properties that correspond to the fields that you want to enable for auditing. For more information, see [Marking a record as relevant](#).

When you enable field-level auditing, your application tracks changes that users make to fields and field groups 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.

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.
3. In the settings panel, click **Auditing**.
4. In the **Auditing** window, turn on the **Enable field audit** switch.
5. In the **Fields** column, select the check boxes next to the fields that you want to enable for auditing.
Note: You can only enable fields, field groups, and field lists that contain scalar properties.
6. If you want to enable fields in a field group, click the name of a field group, and then repeat step 5.
7. Click **Save**.

Result: At run time, when users process the case, they see the changes to the fields on the **Audit** tab. **What to do next:**

- Maintain the good quality of field-level auditing and troubleshoot errors by checking the *pzStandardProcessor* queue processor in Admin Studio.
- Continue working on your field-level auditing data after you deploy your application on another system by packaging the appropriate rules and data. For more information, see [Packaging rules and data that contain field-level auditing information](#).
- [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.

- [Configuring a data model for a case](#)

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](#).

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](#)

Managing concurrent access to a case

Set a locking strategy for parent case types to reduce the risk of losing updates to cases that users make simultaneously.

In most configurations, use the default locking strategy to preserve transaction integrity among cases. If you do not use default locking, you can lose work when other users perform such tasks as bulk processing or escalation actions in a service-level agreement.

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 **Locking**.
Note: Only parent case types support this option.
3. Select one of the available locking strategies.
 - To allow one user at a time to work on a case, click **Allow one user**.

This default strategy locks the case for 30 minutes or until the user submits or closes the case.

Use this approach to enable data updates and to maintain transaction integrity in both parent cases and child cases. For example, the parent case can contain properties that count or total values in the child case. Locking both parent and child cases at the same time helps to keep the counts or totals in sync.

- To allow multiple users at a time to work on a case, click **Allow multiple users**.

This strategy preserves the changes that the first concurrent user makes. All other users who are working on the case receive notifications and must review the changes before they can submit their own updates.

Use this approach if multiple users need to open and review cases simultaneously but do not have to update the cases.

4. **Optional:** To change the default timeout value of 30 minutes, in the **Access time out value is** field, enter a number.
5. Click **Save**.

Result: The locking strategy is applied to new and existing cases. **What to do next:** If you have different concurrent-access requirements for descendants of a parent case, override the locking strategy in the relevant child case types. For more information, see [Overriding the locking strategy of a child case](#).

- [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.

- *Guiding case workers to successful case resolution*
- [Object locking](#)

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.

Before you begin: Create a case type hierarchy. For more information, see [Creating a child case type](#).

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 **Locking**.
3. Customize the locking strategy:

Choices	Actions
Allow users to access a parent case when another user works on a child case	Select the Allow other users to access parent case when the child case is opened check box.
Change the default timeout duration	In the Time out value <i>time-out duration mins.</i> field, enter how many minutes a lock remains on a child case. Tip: Enter a time-out value for the child case that is lower than the time-out value of the parent case.

4. Click **Save**.

- [Managing concurrent access to a case](#)
- [Securing case access](#)

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.

Before you begin: Create a case type. For more information, see [Creating a case type](#). Your application cannot route a temporary case, and only one operator can process and resolve the 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, click the **Settings** tab.
3. In the navigation pane, click **General**.
4. In the **Behavior** section, select the **Create temporary case that is not saved until a 'Persist case' step is reached** check box.
5. Click **Save**.

What to do next: To save a temporary case to the database for reuse in the future, convert the case into a persistent object. For more information, see [Persisting temporary cases](#).

- [Finding duplicate cases](#)

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.

Before you begin:

- Create activity feeds. For more information, see [Creating feed sources for activity feeds](#).
- Ensure that your application includes UI Kit 15 or newer.

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.
3. In the **Pulse** section, configure the Pulse settings:

Choices	Actions
Change the default label	In the Display label field, enter a new label. For example, enter Group chat.
Enable posting Pulse messages	Select the Allow users to add posts check box.
Define a condition for post visibility	In the row of a post type that you want to configure, in the Visibility list, select when the application displays posts.
	a. In the Configure additional feed sources section, click Add Pulse feed . b. In the Pulse feed name list, select the source that you want to add.

Add a feed source	c.	Optional: To display the feed source, default, select the Display feed by default .
Choices	d.	Click OK .

4. Click **Save**.

- [Posting a message in Pulse](#)
- [Enhancing a message in Pulse](#)

Configuring advanced 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.

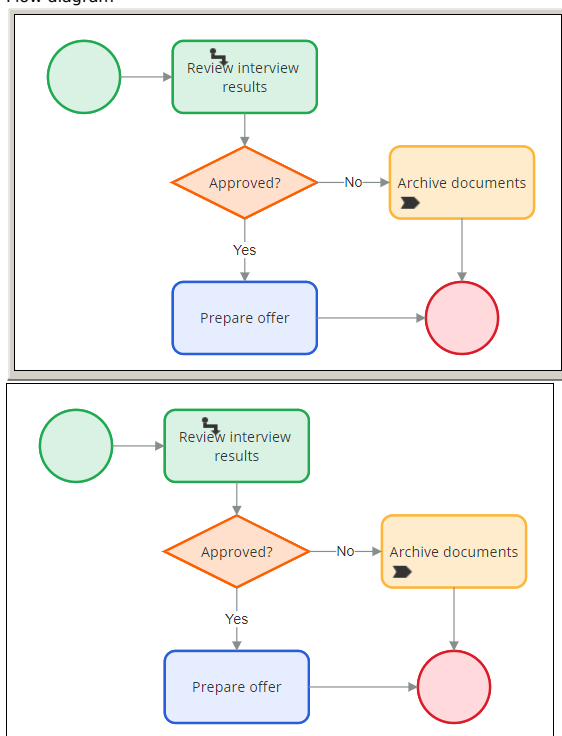
Shapes and connectors

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](#).

For example: The following figure shows a process that visualizes a final part of a recruitment process. After a user, such as a hiring manager, reviews the results of a candidate's job interview, the process reaches a decision point. If the manager approves the candidate, a subprocess for preparing an offer starts. If the manager rejects the candidate, the case moves to a different stage.

Flow diagram



- [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.

- [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.

- [Subprocesses](#)

A subprocess, also called a subflow, is a flow that is called by another flow. You can create a subprocess by adding a step to a process in the life cycle of a case or by adding specific shapes to a flow diagram. Subprocesses modularize the events in the life cycle of a case.

- [Adding a process to a stage](#)
- [Empowering knowledge workers](#)

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.

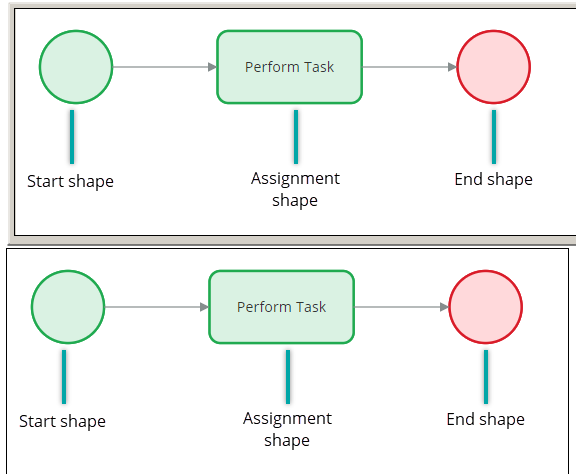
Note: By default, you create processes in Case Designer in a context of a case type, and then you can reuse this process only in this case type. You can create stand-alone processes only in Dev Studio by using a flow diagram. If you want to use a flow diagram in App Studio, you need to add the process to a case life cycle first. For more information, see [Adding a sequential process to a stage](#).

1. Create a rule that stores the process:

- In the header of Dev Studio, click **Create Process Flow** .
- Note:** A flow is a legacy name for a process.
- In the **Label** field, briefly describe the purpose of the process. **For example:** Enter *Collect feedback*.
- Optional:** To change the default identifier for the process, click **Edit**, and then provide a unique value in the **Identifier** field.
- In the **Context** section, select an application layer to store the process.
- In the **Apply to** field, click the Down arrow key, and then select a class that stores the process.
- In the **Add to ruleset** list, select a ruleset and a ruleset version to store the process.
- Click **Create and open**.

Result: 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:

Default process



- 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](#).
- 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.
- Click **Save**.

What to do next: Configure additional processing by performing any of the following actions:

- Add alternate paths to the process. For more information, see [Changing the path of a process](#).
- Supplement your case with a subprocess. For more information, see [Calling one process from another process](#).
- Take advantage of outside functionalities by calling external resources from the process. For more information, see [Integrating a case with an external resource](#).

Flow shapes

A shape in a flow represents a task that a user or application completes as part of a business process. By connecting different types of shapes in a flow, you can define the path that a case follows from creation to resolution.

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.

Unit testing a flow

Flow shapes

A shape in a flow represents a task that a user or application completes as part of a business process. By connecting different types of shapes in a flow, you can define the path that a case follows from creation to resolution.

The following table describes all supported flow shapes. Some shapes are not available for screen flows or flows that you update in the case life cycle instead of the Flow form. **Note:** Flow shapes of the Automations category are available only for flows that belong to a *Work-* class and its subclasses. If you configure a flow that belongs to a *Data-* or *Rule-* class, the number of available automation shapes is restricted.

Name	Description	Icon
Annotation	Adds notes or comments to the flow without affecting how the flow runs.	
Approval	Routes a case to one or more reviewers, based on a user name, reporting structure, or authority matrix.	
Assignment	Creates a task in a work queue or worklist so that a user can provide input to the flow.	
Change Stage	Moves the case to a different stage in the life cycle.	
Create case	Creates a top-level case or one or more child cases.	
Decision	Evaluates an expression or calls a rule, such as a decision tree, to determine which step is next in the flow progression.	
End	Designates the end of flow processing.	
Load data page	Asynchronously starts loading a data page to reduce the time that a screen takes to load the data page.	
Post to Pulse	Creates a message that is sent to the Pulse social stream.	

Name	Description	Icon
Push Notification	Sends a notification to an iOS or Android mobile device to indicate that an action is required by the user.	
Save Data Page	Saves the data in a savable data page to a system of record.	
Send Email	Sends an email to one or more work parties.	
Start	Designates the start of flow processing.	
Subprocess	Calls another flow from the current flow.	
Wait	Pauses an assignment for a specified length of time or until one or more cases reach a specific status.	

- [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.

- [Types of assignments](#)

Each assignment in a flow has a type based on the standard activity that creates the assignment. You select a standard activity when you define the routing options for an assignment in a flow diagram or the life cycle of a case.

- [Annotating a flow](#)

You can use the Annotation shape to add notes and comments to your flow. By documenting the functionality of a flow, you can share information with other developers.

- [Keyboard shortcuts for flows](#)

You can use keyboard shortcuts in the Flow form. Using keyboard shortcuts makes creating flows easier and more efficient.

- [Defining the case life cycle](#)
- [Creating a stand-alone process](#)
- [Creating unit test suites with AUT](#)

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.

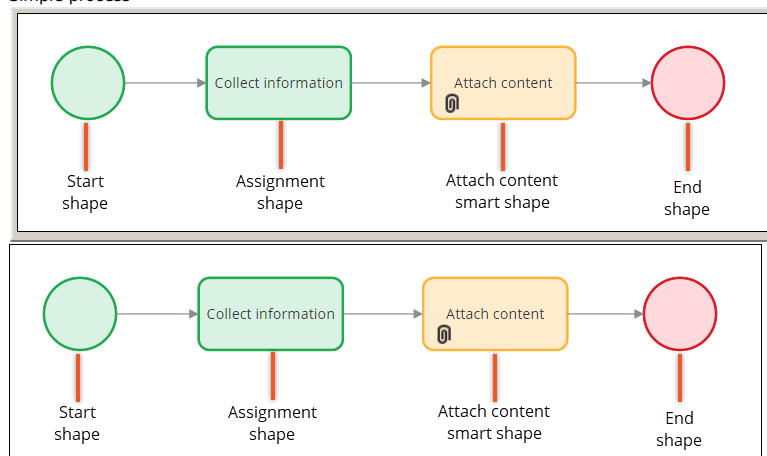
You can build the following types of processes:

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:

Simple process



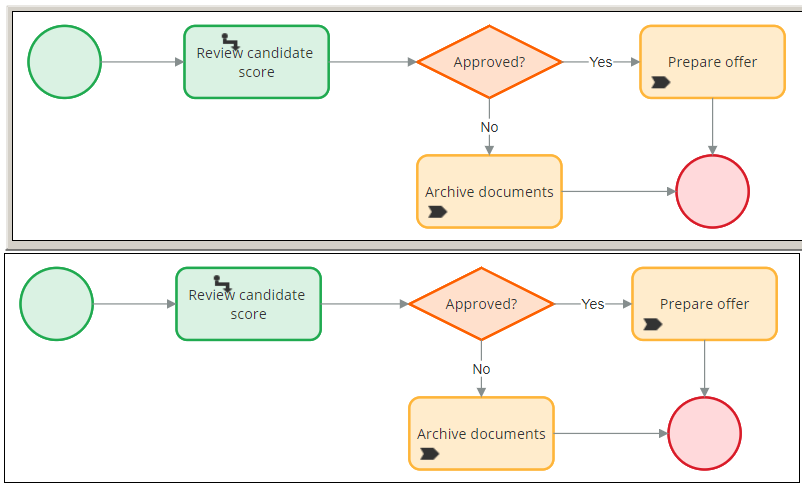
For more information, see [Creating a stand-alone process](#).

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



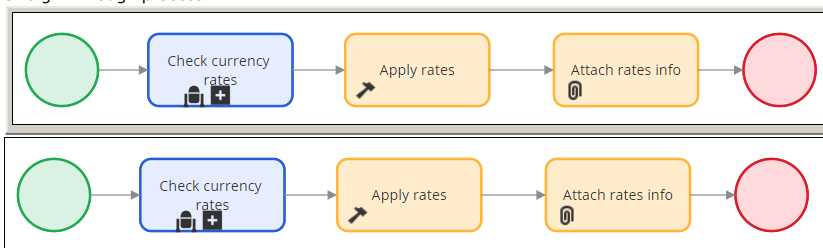
For more information, see [Changing the path of a process](#).

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



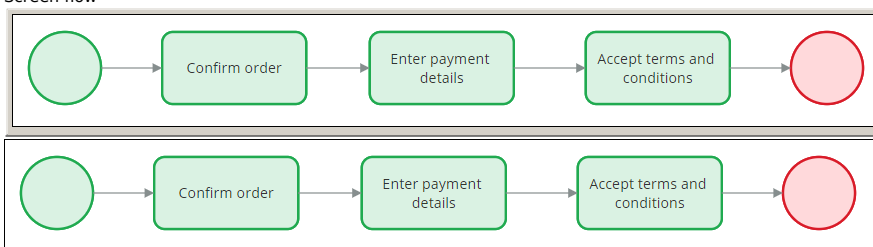
For more information about automations, see [Flow shapes](#).

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](#).

In Case Designer, a counterpart of a screen flow is a multistep form. For more information, see [Adding a multistep form to a stage](#).

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](#).

- Split for Each

For more information, see [Running multiple instances of the same subprocess with a Split for Each shape](#).

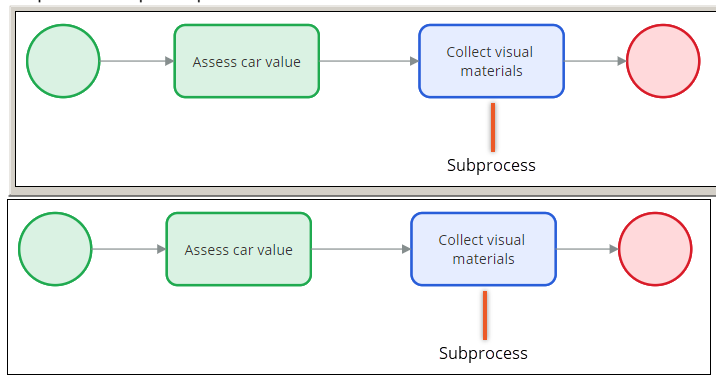
- Split Join

For more information, see [Adding asynchronous processing to a flow](#).

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:

Subprocess in a parent process



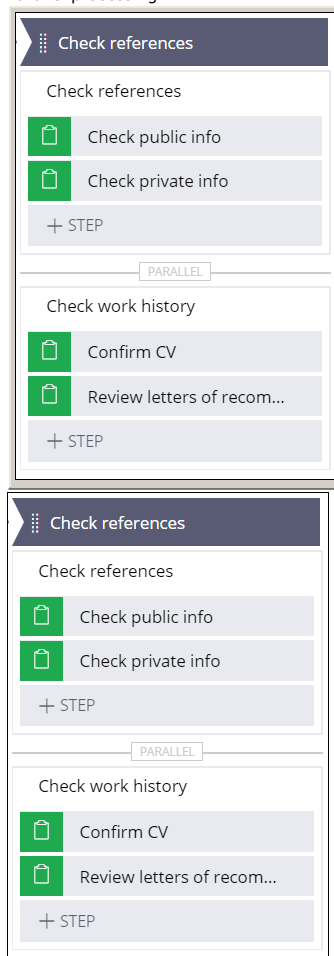
For more information, see [Subprocesses](#).

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](#).

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 [Changing the context of a subprocess](#).

For learning materials about parallel processing, see the [Parallel processing in Pega applications](#) module on Pega Academy.

- [Shape icons in a flow](#)

Shape icons in a flow give a visual indicator of how a shape is configured. You can use shape icons to quickly review the functionality of a flow without having to open the individual shapes.

- [Creating a stand-alone process](#)

Shape icons in a flow

Shape icons in a flow give a visual indicator of how a shape is configured. You can use shape icons to quickly review the functionality of a flow without having to open the individual shapes.

Shapes in a flow support the following icons:

-  — One or more tickets are defined.
-  — A service-level agreement is defined.
-  — The assignment generates one or more notifications.
-  — The assignment is processed by someone other than the current user.
-  — The subprocess is configured as a spin-off flow that runs concurrently with the current flow.
-  — The case status is changed to a value with the `New` prefix when this shape is reached in the flow.
-  — The case status is changed to a value with the `Open` prefix when this shape is reached in the flow.
-  — The case status is changed to a value with the `Pending` prefix when this shape is reached in the flow.
-  — The case status is changed to a value with the `Resolved` prefix when this shape is reached in the flow.
- [Flow shapes](#)
- [Setting a service-level agreement for the Assignment Service shape](#)
- [Responding to business exceptions in a flow](#)
- [Calling one process from another process](#)
- [Changing case statuses](#)

Types of assignments

Each assignment in a flow has a type based on the standard activity that creates the assignment. You select a standard activity when you define the routing options for an assignment in a flow diagram or the life cycle of a case.

Pega Platform supports the following assignment types:

Connect

Creates an assignment that waits for an external system to perform processing before moving to the next step in the flow.

This assignment type supports the Assignment Service shape, which is available only in flow diagrams. For more information, see [Flow shapes](#).

Dependency

Creates an assignment that waits for an event to occur, such as the resolution of another case.

This assignment type is available only in flow diagrams.

Note: As a best practice, use the standard Wait shape instead of a Dependency assignment. 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.

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 for multiple users.

Worklist

Creates an assignment that an application can be route to the worklist of a user.

- [Collecting information from a user](#)
- [Directed Web Access in configuring assignments for external users](#)
- [Pausing and resuming processes in cases](#)
- [Assigning users automatically at run time](#)
- [Assignment shapes in processes](#)

Annotating a flow

You can use the Annotation shape to add notes and comments to your flow. By documenting the functionality of a flow, you can share information with other developers.

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. Hover over a process in a stage, and then click the **Configure process** icon.
4. Click the **Flow shapes** icon, and then click **Annotation**.
5. Double-click the Annotation shape.
6. Replace the default text with your notes or comments.
7. Resize the Annotation shape to fit the text.
8. Drag the Annotation shape to a different position on the canvas, based on the functionality that you are documenting.
9. Click **Save**.

Result:

At run time, annotations do not have any affect on flow processing.

- [Associating a specification with a flow](#)
- [Flow shapes](#)
- [Creating unit test suites with AUT](#)

Keyboard shortcuts for flows

You can use keyboard shortcuts in the Flow form. Using keyboard shortcuts makes creating flows easier and more efficient.

The following keyboard shortcuts are supported.

Shortcut	Description
Shift + mouse click	Adds or removes the selected item from the rest of the selected items.
Delete or Backspace	Deletes the selection.
Ctrl + . *	Toggles the grid visibility.
Ctrl + / *	Toggles the visual guides.
Ctrl + A *	Selects all objects on the canvas.
Alt *	Temporarily uses the pan tool when in select mode or temporarily uses select mode when in pan mode.
Secondary mouse button + drag	Temporarily pans the canvas.
+ (Windows) or = (Mac OS)	Zoom in.

Shortcut	Description
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.

- [Creating unit test suites with AUT](#)

Creating a screen flow

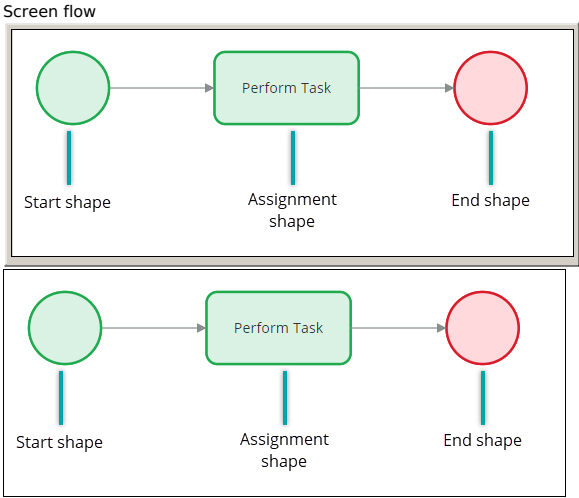
Create a screen flow to walk a single user through a series of screens to collect information. By dividing information into multiple screens, you provide a well-organized and user-friendly interface that is convenient to navigate.

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

Screen flow is a type of process. For more information, about processes, see [Types of processes](#).

- Create the rule that stores the screen flow:
 - In the header of Dev Studio, click **Create Process Flow**.
 - In the **Label** field, describe briefly the purpose of the screen flow. **For example:** Enter Collect appointment booking info.
 - Optional:** To change the default identifier for the screen flow, click **Edit**, and then provide a unique value in the **Identifier** field.
 - Click **View additional configuration options**, and then select **Standard template for screen flows**.
 - In the **Context** section, select an application layer to store the screen flow.
 - 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.
 - In the **Add to ruleset** field, select the name and version of the ruleset that stores the screen flow.
 - Click **Create and open**.

Result: 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:



- Open a dialog box with screen flow configuration by double-clicking the Start shape.
- Optional:** To provide a unique name for the Start shape, in the **Start Shape** field, enter a new value.
- In the **Start** section, define how an application displays the screen flow:

Choices	Actions
Apply a harness that you associate with a case type	In the Harness list, select Use case type policy .
Display navigation to screens as tabs	- In the Harness list, select Custom. - In the Harness name field, press the Down arrow key, and select TabbedScreenFlow7. Result: At run time, an application displays assignments as horizontal tabs, as shown in the following figure: Enter personal details Enter medical history Describe your problem Enter personal details Enter medical history Describe your problem
	- In the Harness list, select Custom. - In the Harness name field, press the Down arrow key, and select TreeNavigation7. Result: At run time, an application displays assignments in a screen flow as a vertical tree layout, as shown in the following figure: Tree layout of a screen flow

Choices	Actions
Display navigation screens as a tree view ☒ Enter personal details ☐ Enter medical history ☐ Describe your problem ☒ Enter personal details ☐ Enter medical history ☐ Describe your problem	
Apply your custom harness	a. In the Harness name field, press the Down arrow key, and select your custom harness. b. In the Harness list, select Custom .

Note: At run time, an application displays every assignment in a screen flow as a separate screen.

- Optional:** To save a case after the last step in a screen flow is complete, select the **Save on last step** check box.
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.
 - Optional:** To allow users to move to the next assignments in a screen flow even if an assignment fails validation, select the **Allow errors** check box.
Note: An option to allow errors is available only after you configure the screen flow to save a case on the last step.
Result: 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.
 - 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. **For example:** Select **Open-InProgress**.
 - 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 user**.
 - To assign the screen flow to another user in your application, select **Specific user**, 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. Click **Submit**.

What to do next: Expand your screen flow with more flow shapes, and then integrate the screen flow with your case life cycle:

- Add shapes to the screen flow to define the different paths that the user can take. For more information about the different shapes that you can use, see [Flow shapes](#).
 - Mark the screen flow as a relevant record, so that you can reuse the flow across your application. For more information, see [Marking a record as relevant](#).
 - Add the screen flow to your case life cycle. For more information, see [Adding a sequential process to a stage](#).
- [Collecting information from a user](#)
 - [Managing work across your team](#)
 - [Assigning tasks to users](#)
 - [About Decision Trees](#)

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:

- [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.
- [Adding a conditional path to a flow](#)

You can use a connector to add a conditional path to a flow. By defining the events that cause your flow to follow different paths, you can create cases that support more than one outcome.
- [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.
- [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.
- [Adding asynchronous processing to a flow](#)

You can use the Split Join shape to call two or more subprocesses that later rejoin the parent flow. By dividing your independent business requirements into multiple processes, you can model different types of dependencies in your flow.
- [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.

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.

Before you begin: Configure a process:

1. Add a process to a stage in your case life cycle. For more information, see [Adding a process to a stage](#).
2. Define the steps for your process. For more information, see [Adding a step to a process](#).
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, hover over a process in a stage, and then click the **Configure process** icon.
4. Above the canvas, click the **Add a flow shape** icon, and then click **Decision**.
5. In the Decision shape, click **[Decision]**, and then replace the default label with a descriptive name. **For example:** In the Decision shape that represents a decision about an approval or rejection, enter *Approved?*.
6. Drag the Decision shape onto the canvas, to a position where the decision occurs in the process.
7. 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.
8. Click the Decision shape.
9. In the Decision shape property panel, define at least one conditional path:

Choices	Actions
Create a custom condition	a. In the When list, select Custom condition . b. Click the Configure conditions icon. c. In the Configure condition window, select a condition, a comparator, and a value to compare with the condition. For example: Case status is equal to New d. Optional: To define more conditions, click the Add row icon, and then repeat substep 9.c . e. Optional: To group the conditions, select comparators from the list. For example: To make only one condition compulsory, from the list, select OR . f. Optional: To reuse the condition in the future, click Actions Add to when conditions library . g. Click Submit .
Use an existing condition	a. In the When list, select Existing condition . b. In the list of conditions, select a condition.

10. In the **Go to** list, select the shape to which the process advances when the case meets the conditions that you define.
11. In the **Otherwise go to** list, select the alternative shape to which the process advances when the case does not meet the conditions.
12. Click **Save**.

Result: 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.

- [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.

- [Changing the decision logic in a flow](#)

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.

- [Adding a parallel process to a stage](#)
- [Conditionally starting a process](#)

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.
- [Adding decisions to processes](#)
- [Changing the decision logic in a flow](#)

Changing the decision logic in a flow

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 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 a process that contains a Decision step, and then click **Open process**.
4. On the **Diagram** tab of the Flow form, double-click the Decision shape to open the property panel.
5. In the **Type** list, select a type of decision logic.

For more information about the supported types, see [Types of decision logic](#).

6. 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:
 1. In the **Rule** field, press the Down Arrow key and select the rule that defines your decision logic.
 2. 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.
 3. 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:
 1. In the **Expression** field, enter a Boolean expression that defines your decision logic.
 2. **Optional:**

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 .

7. Click **Submit** to close the property panel.
8. Click **Save**.

- [Adding decisions to processes](#)
- [Decision tables](#)
- [About Decision Trees](#)
- [Scorecard rules](#)
- [About Predictive Model rules](#)
- [About Map Values](#)
- [Building expressions with the Expression Builder](#)

Adding a conditional path to a flow

You can use a connector to add a conditional path to a flow. By defining the events that cause your flow to follow different paths, you can create cases that support more than one outcome.

1. Open a flow by searching for it or by using the Application Explorer. For more information, see [Finding rules by class](#).
2. Define a new path by connecting a sequence of shapes to an existing shape the flow. For more information about the different shapes that you can use, see [Flow shapes](#).
3. Configure the conditions that cause 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 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](#)
- [Adding a results-based path to a flow](#)
- [Adding decisions to processes](#)

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 flow by searching for it or by using the Application Explorer. For more information, see [Finding rules by class](#).
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](#).
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.
6. Click **Submit**.
7. Click **Save**.

- [Changing the path of a process](#)
- [Collecting information from a user](#)
- [Assignment shapes in processes](#)

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](#).

1. Open a flow by searching for it or by using the Application Explorer. For more information, see [Finding rules by class](#).
2. Define a new path by connecting a sequence of shapes to an existing shape the flow. For more information about the different shapes that you can use, see [Flow shapes](#).
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](#)
- [Creating a screen flow](#)
- [Building forms for case types](#)

Adding asynchronous processing to a flow

You can use the **Split Join** shape to call two or more subprocesses that later rejoin the parent flow. By dividing your independent business requirements into multiple processes, you can model different types of dependencies in your flow.

For example, you can validate tax information and perform a title search as part of a mortgage application flow. Different users can work on the assignments in each subprocess, and the parent flow continues when one or both of the subprocesses return.

Note: Each subprocess runs asynchronously but not in parallel. When a user works on an assignment in one subprocess, the case is locked, which prevents others users from working on assignments in other subprocesses.

1. Add the **Split Join** shape to your flow:
 - a. Open a flow by searching for it or by using the Application Explorer. For more information, see [Finding rules by class](#).
 - b. On the **Diagram** tab, click the **Flow shapes** icon and select **Advanced Split Join**.
 - c. Connect the **Split Join** shape by dragging connector end points to connection points on different shapes in the flow.
2. Double-click the **Split Join** shape to open the property panel.
3. In the **Join** list, select one of the available options to define how the parent flow waits to continue processing .
You can choose from:

All

Waits for all subprocesses to return.

Any

Waits for one subprocess to return. All other subprocesses that were called from the **Split Join** shape stop processing when the parent flow resumes.

Some

Waits for a when condition to return a true value, or a specific number of flows to return a status value that you provide.

4. If the parent flow waits for only some subprocesses to return, define the criteria that must be met before processing continues:
 - To wait for when a condition to return a true value:
 1. In the **Exit iteration** list, select **On when**.
 2. In the **When** field, press the **Down Arrow** key and select the name of a when condition.
 - To wait for a specific number of subprocesses to return:
 1. In the **Exit iteration** list, select **On count**.
 2. In the **Resulting with flow status** field, enter a status value that the parent flow compares with the return value of each subprocess.
 3. In the **Number of paths** field, enter the minimum number of subprocesses that must finish processing and return a value that equals the status in the **Resulting with flow status** field.

For each subprocess that runs when the parent flow reaches the **Split Join** shape:

5. Click + **Add a flow**.
6. Expand the **Specify a flow rule for this subprocess** section.
7. In the **Name** field, enter a clear description of the subprocess.
8. In the **Flow rule** field, press the **Down Arrow** key and select the name of a flow to run.
If the flow supports input parameters, additional fields are displayed for you to define name-value pairs.
9. On the **Subprocess** tab, configure the context of the subprocess.
 - To work on the current case, from the **Define flow** list, select **On current page**.
 - To work on a specific case, from the **Define flow** list, select **On specific work item**. Complete the **Work property**, **Class**, and **Page name** fields.**Note:**
If the case is open as a page on the clipboard, you can leave the **Work property** field blank.
If the case does not already have a page and you do not provide a page name, your application creates a page named `pyNextObj`, `pyNextObj_1`, and so on.
 - To work on an embedded page, from the **Define flow** list, select **On embedded page**. Complete the **Page property** and **Class field** names.**Note:**
The **Page property** field must use a mode of `Page`, `Page List`, or `Page Group`.
If you want to use a class that is not known at design time, in the **Class** field, you must enter a property reference or expression.

When all subprocesses are configured:

10. Close the property panel by clicking **Submit**.
11. Click **Save**.

- [Creating a stand-alone process](#)
- [Changing the path of a process](#)
- [Subprocesses](#)
- [Flow shapes](#)

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](#).
3. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
4. 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**.
5. Activate the ticket in a flow that detects your business exception.

- To automatically activate the ticket:

1. Return to the **Workflow** tab of your case type and click **Life cycle**.
2. 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.

3. On the **Diagram** tab of the Flow form, add the Utility shape to the flow diagram where the exception occurs.
4. 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.
5. Click **Submit**.
6. Click **Save**.

- To manually activate the ticket:

1. [Create a flow action that users perform when the exception occurs](#).

Ensure that the flow action calls the standard *Work-.pyActionSetTicket* activity during its post processing to activate the ticket.

2. Return to the **Workflow** tab of your case type and click **Life cycle**.

3. 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.

4. 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.

5. In the **Advanced** section, click **Local actions**.

6. Click **+ Add local action**.

7. In the **Local Action** field, press the Down Arrow key and select the name of your flow action.

8. Click **Submit**.

9. Click **Save**.

6. **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.

What to do next:

You can test your changes by creating and processing a case that meets the conditions that cause your business exception.

- [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.

- [Creating a ticket](#)

You can create a ticket to help manage business exceptions or errors in a case.

- [Changing the path of a process](#)
- [Adding optional actions to cases](#)
- [Adding optional actions to stages](#)

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.

- [How to implement business exception processing with policy overrides](#)

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:

1. In the **Label** field, enter text that describes the purpose of the ticket.
2. 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.
3. In the **Add to ruleset** field, select the name and unlocked version of a ruleset that stores the ticket.
4. 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](#)
- [Standard tickets](#)
- [Copying a rule or data instance](#)
- [Creating a specialized or circumstance rule](#)

Subprocesses

A subprocess, also called a subflow, is a flow that is called by another flow. You can create a subprocess by adding a step to a process in the life cycle of a case or by adding specific shapes to a flow diagram. Subprocesses modularize the events in the life cycle of a case.

By default, the calling flow waits for the subprocess to finish running before it moves to the next shape in the sequence. You can run a subprocess in parallel with the calling flow, or split it into one or more individual flows that later rejoin. Each subprocess has a context, which might not be the same as the calling flow.

- [Changing the context of a subprocess](#)

You can change the scope of a subprocess that you call from a flow. By running a subprocess in the context of a specific class or case, you can use properties, rules, and other functionality in your application that you cannot access from the parent flow.

- [Calling one process from another process](#)

You can add a subprocess to a process in a stage to call an existing flow in your application. By using subprocesses, you can compartmentalize functionality and simplify the presentation of a process in a case type.

- [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.

- [Calling one process from another process](#)

Changing the context of a subprocess

You can change the scope of a subprocess that you call from a flow. By running a subprocess in the context of a specific class or case, you can use properties, rules, and other functionality in your application that you cannot access from the parent flow.

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 process that contains the subprocess step.
4. Click **Open process**.
5. On the **Diagram** tab of the Flow form, double-click the Subprocess shape to open the property panel.
6. **Optional:** To allow the parent flow to continue processing without waiting for this subprocess to return, select the Spinoff flow check box.
7. Choose a context for the subprocess.
 - To work on the current case, from the **Define flow** list, select **On current page**.
 - To work on a specific case, from the **Define flow** list, select **On specific work item**. Complete the **Work property**, **Class**, and **Page name** fields. **Note:**
If the case is open as a page on the clipboard, you can leave the **Work property** field blank.
If the case does not already have a page and you do not provide a page name, your application creates a page named `pyNextObj`, `pyNextObj_1`, and so on.
 - To work on an embedded page, from the **Define flow** list, select **On embedded page**. Complete the **Page property** and **Class field** names. **Note:**
The **Page property** field must use a mode of Page, Page List, or Page Group.
If you want to use a class that is not known at design time, in the **Class** field, you must enter a property reference or expression.

8. Click **Submit** to close the property panel.
9. Click **Save**.

- [Calling one process from another process](#)
- [Subprocesses](#)

Calling one process from another process

You can add a subprocess to a process in a stage to call an existing flow in your application. By using subprocesses, you can compartmentalize functionality and simplify the presentation of a process in a case type.

By default, subprocesses run in the context of the current case. You can change the context if your subprocess needs to update a specific case or class.

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. Hover over a process in a stage, and then click **+ Step > More > Processes** to display a list of flows.
4. Click the name of a flow, and then review the flow description.
5. Click **Select**.
6. If the flow accepts parameters, pass values for those parameters by entering values in the **Parameters** section of the contextual property panel.
7. Click **Save**.

- [Subprocesses](#)

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. **For example:** To send a case for approval to multiple approvers, enter a page property that stores pages with the details of the approvers. **Result:** 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, 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, you select **Some** and your process resumes after a specified number of subprocesses is complete, configure the conditions to resume processing:

Choices	Actions
Resume processing after a when condition returns a true value	a. In the Exit iteration list, select On when. b. In the When field, enter a when condition that resumes processing after evaluating to true.
Resume processing after a specified number of subprocesses reaches a certain status	a. In the Exit iteration list, select On count. b. In the Number of paths field, enter an integer that determines how many subprocesses need to reach a certain status to resume processing. Note: Enter an integer that is lower or equal to the number of embedded pages in the page property that you provided in step 5. c. In the Resulting with flow status field, enter a status that subprocesses need to reach to resume processing. For example: Enter Resolved-Completed.

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, 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 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. **For example:** To find pages that include the word Pega, enter Pega.
 - b. **Optional:** To require an exact match, select the **Exact match** check box. **For example:** To find pages that include an email address of the Pega domain, enter @pega.com.
 - 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](#)
- [Flow shapes](#)

Modifying case types

Modify your case types to reflect your unique business needs. For example, you can change a case type name, or delete a case type when you no longer need it.

- [Renaming case types](#)

Reflect changes to your business requirements by updating the name of a case type.

- [Reclassifying a case type](#)

Classify an existing case type as a child of another case type to save time and resources.

- [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.

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 **Case type** field, enter a new name for your case type.
3. Click **Save**.

- [Creating a case type](#)

Reclassifying a case type

Classify an existing case type as a child of another case type to save time and resources.

If you classify a top-level case type as a child case type, your application demotes the top-level case type, and then relocates dependencies from the demoted case type to the

new top-level case type in the hierarchy.

1. In the navigation pane, click **Cases** .
2. Hover over a parent case type name and click .
3. Select **Add a child case type**.
4. Click **Existing case type**.
5. Select a case type from the list.
6. Click **Finish**.

- [Creating a case type](#)
- [Creating a child case type](#)

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 header of the Case Types Explorer, click **Manage**.
3. 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** .
4. Click **OK**.

- [Creating a case type](#)

Customizing the Get Next Work logic

You can customize Get Next Work processing to the 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.

Before you begin: Configure stages and steps:

1. Create stages for your case life cycle. For more information, see [Adding a stage to a case life cycle](#).
2. Create steps for the stages. For more information, see [Adding a step to a process](#).

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, define when the case changes the status:

Choices	Actions
Change the status when the case reaches a specific stage	a. Click the header of the stage that you want to configure. b. In the Stage properties pane, on the General tab, in the Set case status on stage entry field, press the Down arrow key, and then select a value that you want to apply.
Change the status when the case reaches a specific step	a. Click the step that you want to configure. b. In the Step properties pane, on the General tab, in the Set case status field, press the Down arrow key, and then select a value that you want to apply.

4. Click **Save**.

Result: At run time, the case changes the status when the case reaches a point that you define in the case life cycle.

- [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.

- [Defining a custom case status](#)

Define a custom case status to more precisely communicate the progress that a case has made toward resolution.

- [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.

- [Defining conditions for skipping a stage](#)

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 Open status to provide more details. For example, you can create an Open-Sales 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 Pending status to specialize the meaning. For example, in a hiring process, you can create a custom Pending-ReferenceCheck 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 Resolved status. For example, in a loan request case, you can create a custom Resolved-LoanGranted status to communicate that the loan request is approved, or a custom Resolved-Rejected 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](#).

- [Standard rules and extension points for case status](#)
- [Assigning tasks to users](#)
- [Assigning users automatically at run time](#)
- [Completing work on time](#)
- [Creating cases](#)
- [Reopening a resolved case](#)

Defining a custom case status

Define a custom case status to more precisely communicate the progress that a case has made 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-.
 - c. In the **Field Name** field, enter **pyStatusWork**.
 - 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](#)
- [Changing case statuses](#)

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](#)
 - [Enabling conditional creation for child cases](#)
 - [Defining a custom case status](#)

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 App Studio, click **Case types**.
3. In the **Case types** panel, click **Options Import case types**.
4. 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.

5. **Optional:** To exclude a case type from the import process, clear the check box next to the name of that case type.
6. Click **Submit**.

Result: 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.

- [Creating a case type](#)
- [Extension points for case-type creation](#)

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.

Theme Theme UI-Kit applications

In the classic Theme Theme UI-Kit applications, with section-based architecture, by default, Pega Platform has two portals for the two most common roles: case manager and case worker. The Case Manager portal displays a case and all the nested subcases and tasks, as well as the associated users and roles. The Case Worker portal allows users to create and process cases that are assigned to them. The Case Manager portal allows users to monitor the case statuses for individual team members or a team queue.

A portal consists of pages that consolidate information for easy access. For example, a portal can include pages for Search, Tags, or Recently accessed items, which allow a user to perform their job more efficiently. The Case Manager portal additionally offers the Dashboard, Reports, and Following standard pages.

My Work page of the Case Worker portal

The screenshot displays the 'My Work' page in the Pega GoGoRoad application. The page is divided into three main sections: a left sidebar, a main content area, and a bottom 'Recents' section.

Left Sidebar:

- New** (with a plus icon)
- My Work** (selected, with a folder icon)
- Pulse** (with a speech bubble icon)
- Spaces** (with a grid icon)
- Documents** (with a document icon)
- My Teams** (with a group of people icon)
- My Cases** (with a folder icon)
- Tags** (with a tag icon)

Main Content Area:

The main content area is titled **My Work** and features a **Refresh list** button. It contains a table with the following columns: **Name**, **Case**, and **Category**.

Name	Case	Category
Review service summary	A-8	Assistance Request
Identify location	A-3	Assistance Request
Select services	C-3	Service
Identify location	A-7	Assistance Request
Identify vehicle	A-6	Assistance Request
Identify vehicle	A-5	Assistance Request
Identify vehicle	A-4	Assistance Request
Enter customer information	A-1	Assistance Request
Enter customer information	A-2	Assistance Request

Bottom Section: Recents

The 'Recents' section shows a list of recently accessed cases, each with a category and a case ID:

- Assistance Request A-3
- Service C-1
- Service C-3
- Assistance Request A-9
- Service C-2

PEGA GoGoRoad

+ New

My Work

Pulse

Spaces

Documents

My Teams

My Cases

Tags

Recents

See all

Assistance Request
A-3

Service
C-1

Service
C-3

Assistance Request
A-9

Service
C-2

My Work

Refresh list

	Name	Case	Category
<1d	Review service summary	A-8	Assistance Request
---	Identify location	A-3	Assistance Request
---	Select services	C-3	Service
---	Identify location	A-7	Assistance Request
---	Identify vehicle	A-6	Assistance Request
---	Identify vehicle	A-5	Assistance Request
---	Identify vehicle	A-4	Assistance Request
---	Enter customer information	A-1	Assistance Request
---	Enter customer information	A-2	Assistance Request

Dashboard page of the Case Manager portal

New Hire Feedback

+ New

Dashboard

My Work

Pulse

Spaces

Documents

My Teams

Reports

Tags

Following

Recents

See all

Employee Engagement...

New Hire Feedback

New Hire Feedback

Apply for passport

New Hire Feedback

Summary for New Hire Feedback

Welcome to
Pega Case Manager

Learn to easily create and manage cases.
To get started, watch an overview of the application, take a tour to familiarize yourself, or get self-directed guidance on Pega Community.
[Take a tour](#) [Open Pega Community](#)

Case stages for New Hire Feedback

Administer Survey (7)

Complete (0)

AS Admin Saravalli

3 OPEN

0 OVERDUE

Product Manager

Name	Case	Category
Question page	E-1001	Employee Engagement Survey
Question page	N-11004	New Hire Feedback
Question page	N-11003	New Hire Feedback

Get next work

Reports

Learn more

Cases by status

Show stages for
New Hire Feedback

7 Cases

Administer Survey Complete

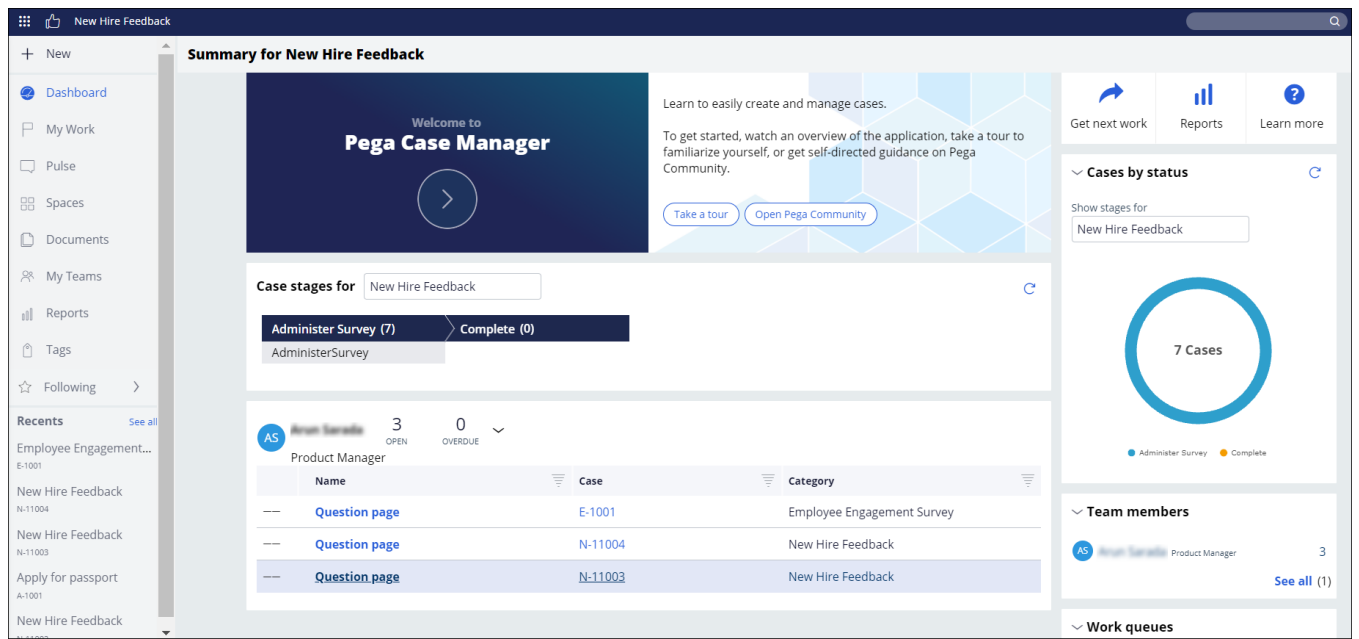
Team members

AS Admin Saravalli Product Manager

3

See all (1)

Work queues



While you can use the default portals as-is in your application, some situations require that you customize the layout of the portal or the tools that you present to users. For more information, see [Building portals](#) and [Organizing the contents of a portal](#).

Note: Pega Platform does not add new features for portals that you build in Theme Theme UI-Kit.

For more information about the types of applications that you can build in Pega Platform, see [Theme UI-Kit](#), [Theme Cosmos](#), and [Cosmos React \(early adopter\)](#).

- [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.
- [Creating cases](#)
Meet your business goals by creating cases that users of your application, such as customer service representatives (CSR) resolve to achieve a specific outcome. When you create a case, you ensure that CSRs can perform their work by collecting and processing information from customers.
- [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 also resolve cases more quickly and effectively.
- [Bulk processing](#)
You can use the Bulk actions form in the Case Manager portal to process more than one case at a time. Bulk processing saves time and is less error prone than processing cases individually.
- [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.
- [Tagging a case](#)
You can associate a tag with a case. A lock is not required to tag a case. You can tag a case regardless of the case's current status.
- [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.
- [Managing case followers](#)
Ensure that relevant stakeholders participate in case discussions by controlling which users follow a case.
- [Configuring your dashboard](#)
Configure your dashboard to personalize your workspace and choose the widgets that increase your productivity.
- [Learning about pages](#)
- [Building portals](#)

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.

Out-of-the-box portal features in the Theme Theme UI-Kit applications

Feature	Description
New	Enables users to start a new case, which is an instance of one of the case types defined in your application and added to the application. For more information, see Creating cases .
	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. You can add Dashboard widgets . You can also share your dashboard with your team. For more information, see Sharing a dashboard . Dashboards are customizable by end users. For more information, see <i>Enabling dashboard personalization for a portal</i> , Configuring dashboards .

Feature

Dashboard
(available only in the Case Manager portal)

New Hire Feedback

+ New

Dashboard

My Work

Pulse

Spaces

Documents

My Teams

Reports

Tags

Following

Recents

Employee Engagement...

New Hire Feedback

New Hire Feedback

Apply for passport

New Hire Feedback

Summary for New Hire Feedback

Welcome to

Pega Case Manager

Learn to easily create and manage cases.

To get started, watch an overview of the application, familiarize yourself, or get self-directed guidance on Community.

Take a tour

Open Pega Community

Case stages for

New Hire Feedback

Administer Survey (7)

Complete (0)

AdministerSurvey

AS

Arjun Saravali

3

OPEN

0

OVERDUE

Product Manager

Name	Case	Category
Question page	E-1001	Employee Engagement Survey
Question page	N-11004	New Hire Feedback
Question page	N-11003	New Hire Feedback

New Hire Feedback

+ New

Dashboard

My Work

Pulse

Spaces

Documents

My Teams

Reports

Tags

Following

Recents

Employee Engagement...

New Hire Feedback

New Hire Feedback

Apply for passport

New Hire Feedback

Summary for New Hire Feedback

Welcome to

Pega Case Manager

Learn to easily create and manage cases.

To get started, watch an overview of the application, familiarize yourself, or get self-directed guidance on Community.

Take a tour

Open Pega Community

Case stages for

New Hire Feedback

Administer Survey (7)

Complete (0)

AdministerSurvey

AS

Arjun Saravali

3

OPEN

0

OVERDUE

Product Manager

Name	Case	Category
Question page	E-1001	Employee Engagement Survey
Question page	N-11004	New Hire Feedback
Question page	N-11003	New Hire Feedback

Feature	Description

Provides quick access to the cases and assignments of users.

My Work view in the Case Worker portal

PEGA

GoGoRoad

+ New

My Work

Pulse

Spaces

Documents

My Teams

My Cases

Tags

Recents

Assistance Request A-3

Service C-1

Service C-3

Assistance Request A-9

Service C-2

See all

My Work

	Name	Case	Category
-1 d	Review service summary	A-8	Assistance Req
---	Identify location	A-3	Assistance Req
---	Select services	C-3	Service
---	Identify location	A-7	Assistance Req
---	Identify vehicle	A-6	Assistance Req
---	Identify vehicle	A-5	Assistance Req
---	Identify vehicle	A-4	Assistance Req
---	Enter customer information	A-1	Assistance Req
---	Enter customer information	A-2	Assistance Req

My Work

Feature		PEGA GoGoRoad Description
	New My Work Pulse Spaces Documents My Teams My Cases Tags Recents See all Assistance Request A-3 Service C-1 Service C-3 Assistance Request A-9 Service C-2 Name Case Category <1d Review service summary A-8 Assistance Req Identify location A-3 Assistance Rec Select services C-3 Service Identify location A-7 Assistance Rec Identify vehicle A-6 Assistance Rec Identify vehicle A-5 Assistance Rec Identify vehicle A-4 Assistance Rec Enter customer information A-1 Assistance Rec Enter customer information A-2 Assistance Rec Displays users' message feed from the Pulse gadget.	
Pulse	Pulse gadget in a case Post AS Start a conversation Post Post AS Start a conversation Post For more information, see Configuring Pulse for case types and Collaborating with users by using	
Spaces	Lists and helps users manage collaboration spaces. For more information, see Collaborating with users by using spaces.	
Documents	Lists application documents that users create or follow. For more information, see Collaborating on shared content by using documents.	
My Teams	Displays the users in your team (a workgroup in Dev Studio) and the number of their open assign For more information, see Creating a team and Configuring teams.	
Reports (available only in the Case Manager portal)	Provides quick access to reports in the application. For more information, see Report Browser features and Editing a report.	
Tags	Lists tags added by application users. For more information, see Tagging a case and Find cases faster with improved case tags (8.4	
Following (available		

only in the Feature Case Manager portal)	Displays a responsive list of work items that you currently follow.
	For more information, see Following a case .
My Cases (available only in the Case Worker portal)	Displays all the open cases to which you are a party.

For more information, see [Generating the navigation menu for a portal](#), [Organizing the contents of a portal](#), [Creating the main menu for a portal](#), [Editing a page of an application](#) and [Portals and pages](#).

- [Learning about pages](#)
- [Building portals](#)

Creating cases

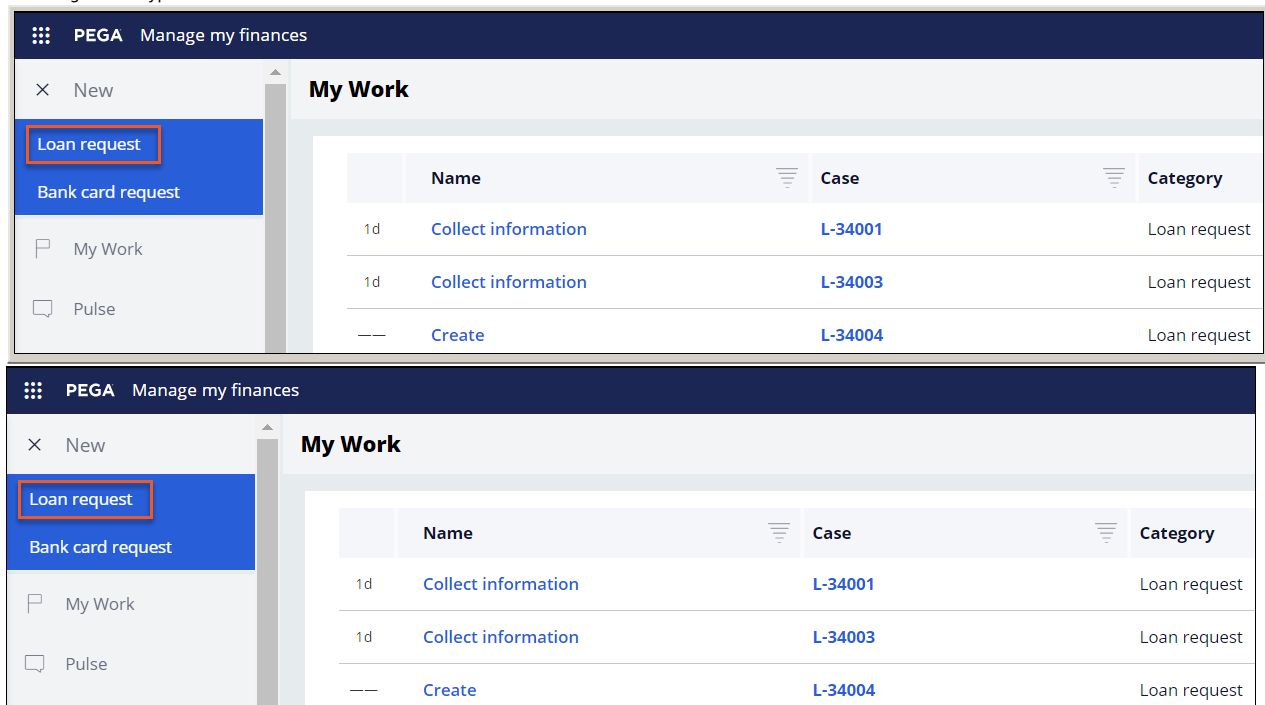
Meet your business goals by creating cases that users of your application, such as customer service representatives (CSR) resolve to achieve a specific outcome. When you create a case, you ensure that CSRs can perform their work by collecting and processing information from customers.

To create a case, you use a case type, which is a template of one of your business processes. Case types consist of stages, processes, and steps, that the case worker completes to resolve the case.

Before you begin:

1. Create case types that you can reuse for multiple business processes. For more information, see [Creating a case type](#).
2. Define which cases the user can create. For more information, see [Creating the main menu for a portal](#).
1. In the header of App Studio, click the name of a portal in which you process cases.
2. In the navigation panel, click **New**, and then select the case type that you want to use to create your new case. **For example:** The following figure shows creating a case in the Case Worker portal.

Selecting a case type to create a case.



Result: You have created a case, which can now be processed to achieve a specific result.

- [Creating a top-level case type](#)
- [Defining the case life cycle](#)
- [Configuring a case at run time](#)

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 also resolve cases more quickly and effectively.

Before you begin: Enable transfer of assignments for your process. For more information, see [Enabling users to transfer an assignment](#).

The `pyTransferAssignment` flow action that is present on the assignment shape in your process enables users to manually reassign an assignment to another operator, a reporting manager, work queue, work group or work group manager. After a transfer, the assignment appears in the worklist of the new assignee.

1. Log in to an end-user portal, for example the Case Manager 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, in the upper-right corner, click **Actions > Transfer assignment**.
4. In the **Activities** pane of the case, in the **Transfer assignment** list, select the new assignee. For example, select **User**, and then enter the user name.
5. Click **Transfer**.

- [Enabling users to transfer an assignment](#)

By adding the `pyTransferAssignment` flow action to the assignment shape in your process, you enable end users to transfer assignments at run time, which can speed up case processing and support selection of the most competent worker to resolve a particular case.

Enabling users to transfer an assignment

By adding the *pyTransferAssignment* flow action to the assignment shape in your process, you enable end users to transfer assignments at run time, which can speed up case processing and support selection of the most competent worker to resolve a particular case.

The *pyTransferAssignment* flow action enables users to manually reassign an assignment to another operator, a reporting manager, work queue, work group, or work group manager. After a transfer, the assignment appears in the worklist of the new assignee.

1. In the navigation pane of Dev Studio, click **App**.
2. On the **Classes** tab, expand the case type in which you want to edit the assignment, and then click **Process Flow**.
3. Click the name of the flow that you want to modify.
Note: You can modify any flow apart from the flow for the Create stage.
4. In the **Flow** form, on the **Diagram** tab, double-click an assignment.
5. In the **Assignment properties** dialog box, in the **Advanced** section, in the **Local action** list, select **pyTransferAssignment**.
6. Click **Submit**.
7. Click **Save**.

Bulk processing

You can use the Bulk actions form in the Case Manager portal to process more than one case at a time. Bulk processing saves time and is less error prone than processing cases individually.

You can perform the following tasks in bulk:

- Creating a case
- Updating case information
- Running a flow action to update a case
- [Creating cases in bulk](#)

You can create multiple cases at once by using the Bulk actions form. Create cases in bulk to define case details, such as labels and work statuses, in one step without manually interacting with individual forms.

- [Updating cases in bulk](#)

You can update multiple cases at once by using the Bulk actions form. Update cases in bulk to set the values of case details, such as labels and descriptions, in one step without manually interacting with individual forms.

- [Running flow actions in bulk](#)

You can run the same flow action for multiple cases at once by using the Bulk actions form.

- [Troubleshooting errors in bulk processing](#)

The following issues can arise when you bulk process cases in your application:

Creating cases in bulk

You can create multiple cases at once by using the Bulk actions form. Create cases in bulk to define case details, such as labels and work statuses, in one step without manually interacting with individual forms.

Note: To define more case details during bulk creation, customize the *pyBulkCreate* section by adding relevant properties to the grid.

To create cases in bulk:

1. Ensure that your access group is associated with a role that grants you the *Work-.PerformBulk* privilege.
2. In the Case Manager or Case Worker portal, select **Bulk actions** from the operator menu.
3. Click **Create**.
4. In the **Case Type** list, select the name of a top-level case type.

Bulk creation is not supported for child case types.

5. Click **+ Add Case** and enter values in the fields that are displayed.

The default fields include:

- **Label** — The name of the case that is visible in worklists and forms.
Labels are stored in the *Work-.pyLabel* property.
- **Description** — The purpose of the case.
Descriptions are stored in the *Work-.pyDescription* property.
- **Status** — An initial indication of the progress that a case has made towards resolution.
Statuses are stored in the *Work-.pyStatusWork* property and are automatically updated as case workers process a case.

6. Repeat step 5 for each case to create.

7. Click **Create Cases**.

Cases are created based on the values that you provide and a status is displayed in the **Processing Results** column of the case list.

- [Bulk processing](#)
- [Troubleshooting errors in bulk processing](#)

Updating cases in bulk

You can update multiple cases at once by using the Bulk actions form. Update cases in bulk to set the values of case details, such as labels and descriptions, in one step without manually interacting with individual forms.

Tip: To set more case details during a bulk update, customize the *pyBulkUpdate* section by adding relevant properties to the grid.

To update cases in bulk:

1. Ensure that your access group is associated with a role that grants you the *Work-.PerformBulk* privilege.
2. To bypass validation logic, add a role to your access group that grants you the *pxIgnoreBulkValidations* privilege.
Bypassing validation logic allows you to reopen or update resolved cases and ignore the validation criteria that are defined in the case type.
3. In the Case Manager or Case Worker portal, select **Bulk actions** from the operator menu.
4. Click **Update**.

5. In the **Case Type** list, select the name of a case type to display its cases.
6. Add filter criteria to refine the list of cases.
 1. Click **+ Add Filter**.
 2. In the **Filter Property** field, enter the name of a property to evaluate in each case.
 3. Select a comparator from the **Condition** list.
 4. In the **Filter Value** field, enter a case-sensitive value that is compared with the value of the **Filter Property**.
For example, you can find cases with an urgency that is greater than 80.
 5. Repeat steps a through d to define multiple filters.
 6. Click **Filter Cases** to display cases that meet all the filter criteria.
7. For each case in the list, update the values in the **Name** or **Description** column.
8. Click **Update**.

Cases are updated based on the values that you provide and a status is displayed in the **Processing Results** column of the case list.

- [Validating case data](#)
- [Bulk processing](#)
- [Troubleshooting errors in bulk processing](#)

Running flow actions in bulk

You can run the same flow action for multiple cases at once by using the Bulk actions form.

Run flow actions in bulk to work on cases without impacting their path toward resolution. For example, you can extend a service-level agreement or transfer a set of assignments to a user.

To run a flow action in bulk:

1. Ensure that your access group is associated with a role that grants you the *Work-.PerformBulk* privilege.
2. In the Case Manager or Case Worker portal, select **Bulk actions** from the operator menu.
3. Click **Process**.
4. Modify the filter criteria to refine the list of cases.

By default, the filter criteria return a list of cases with at least one open assignment that is assigned to you.

1. Click **+ Add Filter**.
2. In the **Filter Property** field, enter the name of a property to evaluate in each case.
3. Select a comparator from the **Condition** list.
4. In the **Filter Value** field, enter a case-sensitive value that is compared with the value of the **Filter Property**.

For example, you can find cases that were recently created by comparing the **Create Date/Time** property with a specific date.

5. Repeat steps a through d to add more filters.
6. Click **Filter Cases** to narrow the list of results.
5. Select the check box next to each case for which the flow action will be run.
6. Click **Select Action** and select the flow action to run.

The list of flow actions can vary because it is populated by the intersection of flow actions that are supported by the cases that meet your filter criteria.

7. In the dialog box that opens, enter values in the fields that are displayed and click **OK** to submit the form.

Note: Some flow actions support background processing. Select the **Run in background** check box when you are running a flow action for many cases. An agent queues each request and sends you an email when processing is finished.

The flow action is run for each case, based on the values that you provide and a status is displayed in the **Processing Results** column of the case list.

- [Bulk processing](#)
- [Pega-ProCom agents](#)
- [Troubleshooting errors in bulk processing](#)

Troubleshooting errors in bulk processing

The following issues can arise when you bulk process cases in your application:

- The Bulk actions option is not displayed in the operator menu of your portal.
 1. Associate a role with your access group that grants you the *Work-.PerformBulk* privilege.
 2. Refresh your session.
- The list of items to bulk process is read-only.
 1. Hover over the **Lock** icon in the list 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.
You cannot update a case while it is locked by another user.
 3. After the user has closed the case, click **Update** to refresh the list.
- The **Processing Results** column displays a failed status.
 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.
- The **Processing Results** column never displays a status.
 1. Use the Records Explorer to view the list of agents in your application.

2. Verify that the *Pega-ProCom* agent exists on your node.
 3. Open the Agents form for the *Pega-ProCom* agent.
 4. On the **Schedule** tab, add the *AgentBulkProcessing* activity to the Scheduled Agents list and select the **Enabled** check box.
 5. Select the **Enable this agent** check box.
 6. Save the Agents form.
- The **Instruction** column displays errors.
 1. Click **Dev Studio > Case Management > Tools > Work Admin > Worklist Assignment Errors**.
 2. Review the list of problem assignments and their corresponding error messages to determine which errors can be cleared.
 - You are not notified when background processing is completed.
 1. Ensure that the *Default* email account is correctly configured in your application.
 2. Open your Operator form and verify that a valid email address is provided in the **Email** field.
 - The list of flow actions in the **Select Action** list is incomplete.

The **Select Action** list displays only flow actions that are visible to your current workpool and have a common ancestor class to the cases that you are bulk processing.

To display a flow action that is disqualified from bulk processing:

1. Open the flow action that is missing from the list by using the Records Explorer or searching for it.
2. On the **Action** tab of the Flow Action form, clear the **Disqualify this action from bulk processing** check box.
3. Save the Flow Action form.

To display flow actions for cases:

1. On the Bulk actions form, remove filter criteria that evaluates an assignment in the **Filter Property** column.
For example, delete rows that contain Assigned to Operator or Assigned to Work queue.
2. Click **Filter Cases**.
3. Select the check box next to each case for which the flow action will be run.

To display flow actions and connector actions for assignments:

1. On the Bulk actions form, add filter criteria that evaluates an assignment in the **Filter Property** column.
For example, add rows that contain Assigned to Operator or Assigned to Work queue.
2. Click **Filter Cases**.
3. Select the check box next to each assignment for which the flow action will be run.

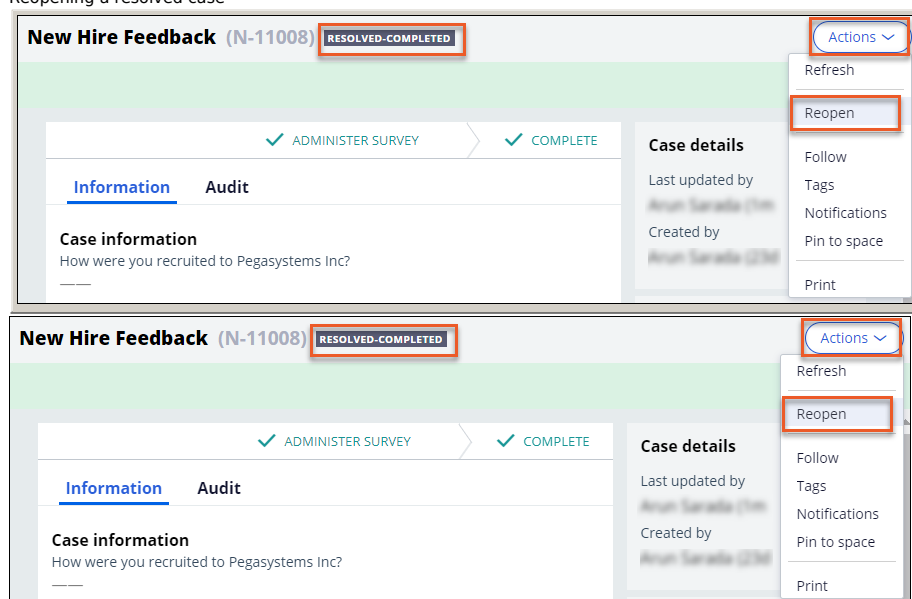
- [Creating cases in bulk](#)
- [Updating cases in bulk](#)
- [Running flow actions in bulk](#)

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 the Case Manager portal.
2. Find and 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**. For example:

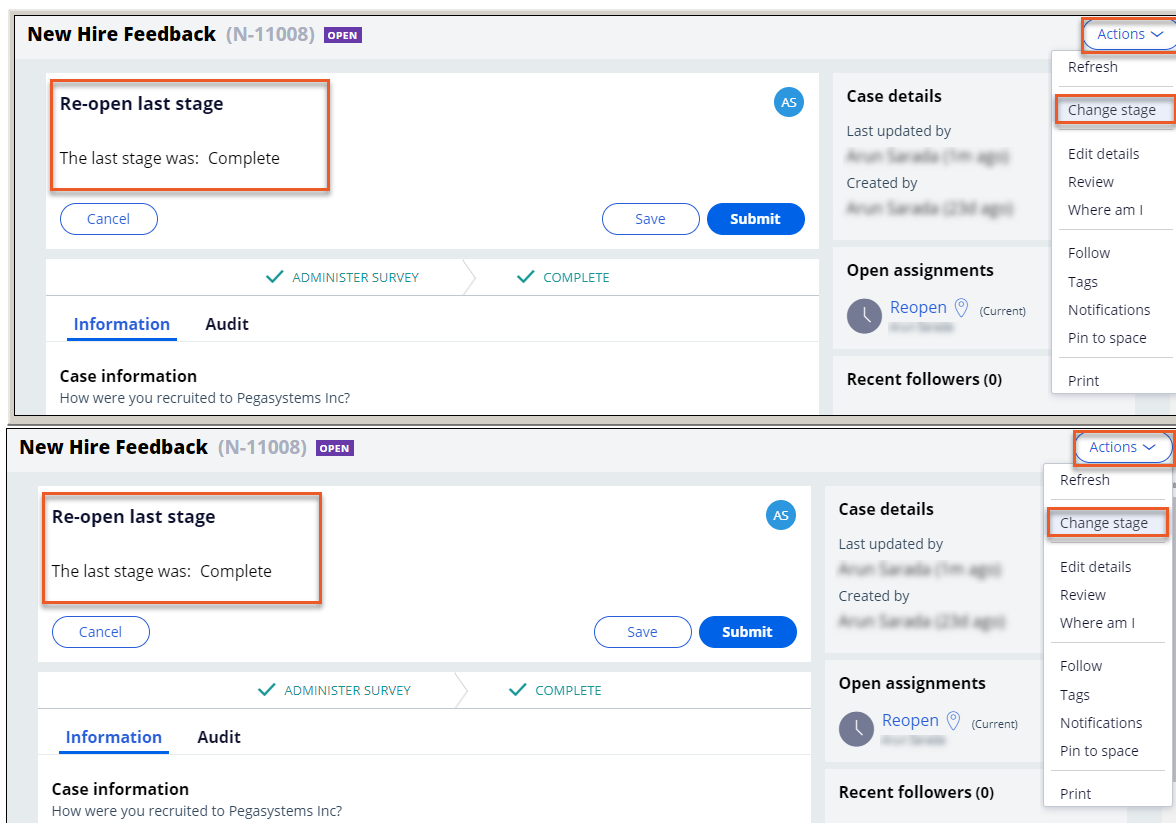
Reopening a resolved case



Result: The case status changes to **Open** and the process that is defined in the *pyFlowName* property restarts. If the case type does not use stages, you can see the case review form. If the case type uses stages, the last stage reopens, which you can move to any stage.

For example, the following figure shows a case with stages: that is reopen to the last stage; the case you can move to a different stage:

A reopened case showing the last stage with an option to change to a different stage



- [Changing case statuses](#)

Tagging a case

You can associate a tag with a case. A lock is not required to tag a case. You can tag a case regardless of the case's current status.

1. Open a case by searching for it or clicking a case ID in your worklist.
2. Access the case and click the **Tags** icon. The Manage Tags dialog displays.
3. Select from the list of existing tags or type a new tag, and then click the **Add Tag** button. The tag appears in the Associated Tags list.

To delete a tag from a case, click the **Delete** button associated with the tag you want to remove. This deletes the link from the tag to the case; it does not delete the tag from the database table. See [Pega-EndUserUI agent](#).

- [Enabling case tags](#)

Speed up case search by enabling case tags. When users tag cases, they can filter cases and Pulse messages to retrieve relevant information.

- [Creating suggested tags for cases](#)

Ensure that users can find relevant cases by providing suggested tags. When users apply consistent tagging in cases and Pulse messages, they faster obtain information that they need.

- [Tagging cases](#)
- [Creating suggested tags for cases](#)

Enabling case tags

Speed up case search by enabling case tags. When users tag cases, they can filter cases and Pulse messages to retrieve relevant information.

Users can filter cases and messages 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.

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.
3. In the settings navigation panel, click **Collaboration**.
4. In the **Tags** section, select the **Enable tags** check box.

Result: At run time, users can add tags directly in the case view. **What to do next:** To improve case search, ensure that users apply consistent tagging by creating suggested tags. For more information, see [Creating suggested tags for cases](#).

- [Collaborating with users by using Pulse](#)

Creating suggested tags for cases

Ensure that users can find relevant cases by providing suggested tags. When users apply consistent tagging in cases and Pulse messages, they faster obtain information that they need.

Define suggested tags so that users can select them from a list when they tag a case instead of typing tags.

Before you begin: Enable case tags. For more information, see [Enabling case tags](#).

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.
3. In the settings navigation panel, click **Collaboration**.
4. In the **Tags** section, in the **Configure tags** window, enter a tag that you want to create as a suggested tag, and then click **Add**. **For example:** case_update OR finances.
5. Click **Save**.

Result: At run time, when users tag a case, they can select suggested tags instead of typing them.

- [Tagging cases](#)

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.

Before you begin: Ensure that you set your user preferences to control the frequency at which you receive notifications from the cases that you follow. For more information, see [Setting notification preferences](#).

1. Log in to an end-user portal, for example, the Case Manager 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, in the upper-right corner, click **Actions > Follow**.
4. **Optional:** To stop following the case, click **Unfollow**.

Result: You now follow a case and the utilities pane displays you as a follower of the case.

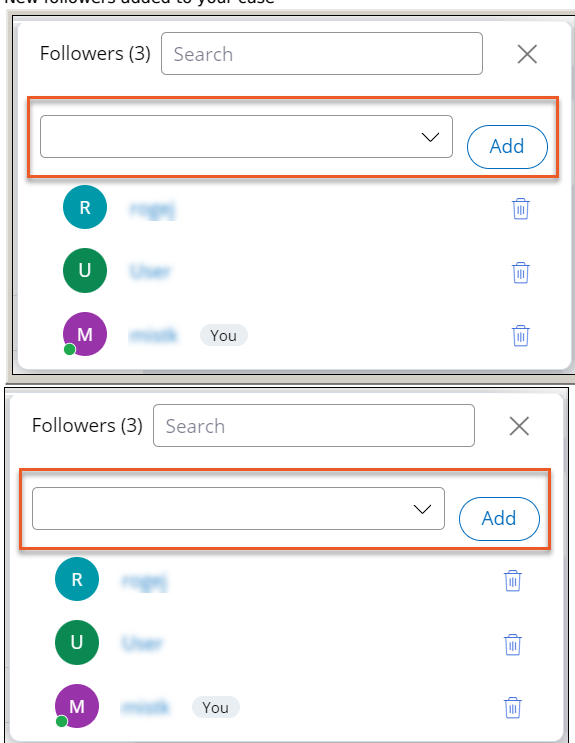
- [Collaborating with users by using Pulse](#)

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, the Case Manager 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 **Recent Followers** section, click the **Gear** icon.
4. In the **Followers** dialog box, in the drop-one or more users in your application, and then click **Add**. **For example:**

New followers added to your case



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.

Result: Case followers receive notifications for the case, based on the settings in their notification preferences.

- [Following a case](#)
- [Setting notification preferences](#)

Configuring your dashboard

Configure your dashboard to personalize your workspace and choose the widgets that increase your productivity.

Before you begin:

- Your application must have personalization enabled. For more information, contact a developer on your team who has access to Dev Studio.
- You must have the *pxDashboard* privilege.
- [Changing your dashboard layout](#)

Find basic information about your application more efficiently by organizing your dashboard in a comprehensive and transparent way.

- [Adding a widget to your dashboard](#)

Increase the productivity of your team by personalizing the dashboard with widgets that display relevant information. For example, you can design a workspace that includes a widget for frequently used reports to help managers save time and oversee their projects.

- [Sharing a dashboard with your team](#)

As a team manager, you can publish your dashboard to share it with your team. By providing a baseline layout with commonly used tools, you can help team members

save time while they set up their environment.

- [Dashboard widgets](#)
- [Sharing a dashboard with your team](#)

Changing your dashboard layout

Find basic information about your application more efficiently by organizing your dashboard in a comprehensive and transparent way.

1. In the header of your workspace, click the **Personalize the dashboard** icon.
2. In the **Edit dashboard** panel, click **Switch template**.
3. In the **Select a template** window, review the template preview, and then select the template that you want to use. The template previews include the numbers, sizes, and relative positions of slots that they contain.
4. Click **Publish**.

Result: Users see an updated version of the dashboard when they log on to your application.

- [Sharing a dashboard with your team](#)

Adding a widget to your dashboard

Applicable to **Theme UI-Kit** applications

Increase the productivity of your team by personalizing the dashboard with widgets that display relevant information. 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**.
 - b. In the **Add widgets to slot** window, select one or more widgets that you want to add to the slot, and then click **Add selected**. **Result:** The widget appears at the bottom of the slot.
 - c. **Optional:** To change the position of the widget, drag the widget into the correct 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.

You can also add or delete widgets by using the **Add widget(s)** and **Delete this widget** buttons in the user interface preview.

4. In the work area, hover over the newly added widget, and then determine if there are additional options to configure:

Choices	Actions
Click to edit	a. Click the widget. b. In the Edit dashboard pane, edit the widget with the configuration controls. c. Click Save. d. Click Publish.
No Click to edit message	Click Publish .

- [Dashboard widgets](#)
- [Sharing a dashboard with your team](#)

Sharing a dashboard with your team

As a team manager, you can publish your dashboard to share it with your team. By providing a baseline layout with commonly used tools, you can help team members save time while they set up their environment.

Before you begin:

- Verify that your team has not personalized their dashboards yet, because the changes that you publish do not override or reset personalized dashboards.
- Add commonly used widgets to your dashboard. For more information, see [Configuring your dashboard](#).

1. In the header of your workspace, click the **Personalize the dashboard** icon.
2. In the **Edit dashboard** panel, click **Publish Publish to team**.

- *Setting up your environment*

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.

When you define a goal for an assignment, you define the suggested time required to resolve the assignment. When you define a deadline, you define the ultimate time in which to resolve the assignment. You can also define escalation actions that your application performs when the goal and deadline expire. For example, if a case worker does not resolve their task by a week, that case worker receives a notification reminding them to do so. If the case worker does not resolve the task for another week, the manager of that case worker receives a notification.

- [Learning about service-level agreements \(SLAs\)](#)

Standardize the way that case workers resolve cases in your application by enforcing service-level agreements (SLAs). By implementing SLAs, you define a timeline of delivering the product and make a commitment to your customers.

- [Urgency](#)

Urgency is a numeric value, ranging from 0 to 100, that is displayed in worklists and cases. You can use urgency to bring visibility to unresolved assignments and cases in your application.

- [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. SLA measurement starts when the case starts and ends when the case finishes.

- [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.

- [Setting goals and deadlines on the Service-level agreement \(SLA\) form](#)

Ensure that your cases are resolved on time by defining goals, deadlines, and passed deadline actions. Complement the time intervals with increasing urgency and escalation actions.

- [Creating a service-level agreement \(SLA\) rule](#)

Enforce service-level agreements (SLAs) and maintain timely resolution of cases by defining goal, deadline, and passed deadline time intervals. By creating a Service-level Agreement (SLA) rule, you can quickly reuse the SLA settings for multiple assignments.

- [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.

- [Defining an escalation action for a stage and a process](#)

Ensure that your business 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.

- [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.

- [Automating work by creating case types](#)

Learning about service-level agreements (SLAs)

Standardize the way that case workers resolve cases in your application by enforcing service-level agreements (SLAs). By implementing SLAs, you define a timeline of delivering the product and make a commitment to your customers.

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.

An 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. The following types of SLAs are supported:

Case

Starts when a user starts or reopens a case, and stops when they resolve the case.

Stage

Starts when a case enters the stage, and ends when the case leaves the stage.

Process in a stage

Starts when a process starts, and ends when the last step in the process finishes.

You can create processes, which are flows, by defining a life cycle for a case. You can also create a process as an optional action.

Assignment

Starts when the user or an automation creates or routes assignment to a work queue or worklist, and ends when the assignment finishes or stops due to an error condition.

The time at which a user opens the form for an assignment has no effect on the SLA.

Approval step

Starts when the subprocess for the Approval shape starts, and ends when the subprocess reaches an End shape.

At run time, an agent detects unmet goals and deadlines in your application. You can define escalation actions, such as notifying a work party, for this agent to run to help users resolve cases faster.

Urgency

Urgency is a numeric value, ranging from 0 to 100, that is displayed in worklists and cases. You can use urgency to bring visibility to unresolved assignments and cases in your application.

The higher the urgency, the more important it is to address the unresolved item. You can adjust the urgency of an item by using a service-level agreement or by using the following standard rules:

- *Work-.UrgencyUpdate* — A flow action that you can call while processing a case.

The integer that you provide is added 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 value that you provide is evaluated by the *Work-.pxUrgencyWork* declare expression.

- *Assign-.pyUrgencyAssignAdjust* — A property that you can set to increase or decrease the current urgency of an assignment.

The value that you provide is evaluated by the *Assign-.pxUrgencyAssign* declare expression.

At run time, urgency is updated each time that the value of a standard property changes or a user saves a case.

- [Completing work on time](#)
- [Setting goals and deadlines for case resolution](#)
- [Setting goals and deadlines for assignment completion](#)

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. 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.

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 **Goal & deadline**.
3. In the **Use Service-level agreement (SLA)** list, define the SLA criteria:
 - To define custom criteria, select **Custom SLA**, and then go to step [4](#).
 - To reuse criteria from an existing service-level agreement, select **Existing SLA**. In the auto-complete field, enter the service-level agreement name, and then go to step [9](#).
 - To leave a case without an SLA, select **Never**, and then go to step [9](#).
4. 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**.
5. In the **Goal** section, in the **Days** and **HH:MM:SS** fields, enter the suggested resolution time for a case. **For example:** To indicate that the goal is a day, 6 hours, and 30 minutes, in the **Days** field, enter 1, and then, in the **HH:MM:SS** field, enter 06:30:00.
6. 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, in the **Days** and **HH:MM:SS** fields, enter the required resolution time for a case.
8. In the **Increase urgency by** field, enter a value by which you want to increase the current case urgency when the deadline elapses.
9. Click **Save**.

Result: 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.

- [Setting service-level agreements \(SLAs\) for stages, processes, and steps](#)
- [Defining an escalation action for a case](#)

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.

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](#).

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.
 - 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 property 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 5.
 - 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.
6. In the **Goal** section, define custom SLA criteria for the goal:
 - a. In the **Goal** section, in the **Days** and **HH:MM:SS** fields, enter the suggested resolution time for a case. **For example:** To indicate that the goal is 1 day, 6 hours, and 30 minutes, in the **Days** field, enter 1, and then, in the **HH:MM:SS** field, enter 06:30:00.
 - b. 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:
 - a. In the **Deadline** section, in the **Days** and **HH:MM:SS** fields, enter the suggested resolution time for a case. **For example:** To indicate that the deadline is a day, 6 hours, and 30 minutes, in the **Days** field, enter 1, and then, in the **HH:MM:SS** field, enter 06:30:00.
 - b. 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**.

Result: 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. **What to do next:** Notify stakeholders about an elapsed goal or deadline. For more information, see [Defining an escalation action for a stage and a process](#).

- [Setting service-level agreements \(SLAs\) for case resolution](#)

Setting goals and deadlines on the Service-level agreement (SLA) form

Ensure that your cases are resolved on time by defining goals, deadlines, and passed deadline actions. Complement the time intervals with increasing urgency and escalation actions.

By using the Service-level agreement (SLA) form, you can conveniently define advanced SLA options on one form, and then you can reuse your settings for multiple cases.

Tip: As a best practice, use case-type settings to set goals and deadlines because they automatically manage and synchronize the details of your SLA.

1. To ensure that rules managed by your application do not conflict with your SLA rule, on the **Goal & deadline** tab, from the **Use Service-level agreement (SLA)** list, select **Never**.
2. Open an SLA rule that is in the class path of your case type by clicking **Records Process Service Level Agreement**, and then performing one of the following actions:
 - Select a rule that you want to edit.
 - Create a new rule by clicking **Create** on the **Service Level Agreement** tab.For more information about creating a rule, see [Creating a rule](#).
3. 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 a default urgency of 10. The system adds the value in this field to arrive at the initial urgency. For example, if you enter 10, the initial urgency is 20.
4. From the **Assignment ready** list, select an option to define the starting time for the assignment:
 - To send the assignment to a worklist or a work queue immediately after it is created, select **Immediately**.
 - To define the starting time by using a *DateTime* 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 specific period of time, select **Timed delay**, and then in the **Days**, **Hours**, and **Minutes** fields, specify the delay.
5. In the **Calculate service levels** field, select a method of calculating intervals for goal and deadline:
 - To specify the intervals manually, select **Interval from when assignment is ready**, and then in the **Goal** and **Deadline** sections, in the **Days**, **Hours**, and **Minutes** fields, 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 times.
6. **Optional:** In the **Passed deadline** section, define the number of the passed deadline events and the time when the passed deadline events occur.
7. In the **Amount to increase urgency** field, enter the amount to increase the urgency when the time interval passes. The maximum urgency for the assignment is 100, which is the sum of the initial and increased urgencies.
8. **Optional:** Define an escalation action that occurs after goal, deadline, or passed deadline elapses. For more information, see *Defining an escalation action for an unresolved case*.
9. Click **Save**.
10. Open the *pyDefault* data transform in the class of your case type by clicking **Records Data Model Data Transform**. The data transform runs each time that a case is created.
11. Set the value of the *pySLAName* property to the name of your SLA rule.
12. Click **Save**.

Result:

At run time, an agent and the *Work-.Overall/SLA* flow monitor the SLA, and the Case Details section on a case displays how much time remains to meet the goal or deadline.

- [Goals and deadlines](#)
- [Creating unit test suites with AUT](#)
- [Defining an escalation action for an unresolved case](#)

Creating a service-level agreement (SLA) rule

Enforce service-level agreements (SLAs) and maintain timely resolution of cases by defining goal, deadline, and passed deadline time intervals. By creating a *Service-level Agreement* (SLA) rule, you can quickly reuse the SLA settings for multiple assignments.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Service Level Agreement**.
3. On the **Service Level Agreement** tab, click **Create**.
4. In the **Label** field, enter a short description for the rule.
5. In the **Context** section, select an application to which the rule applies.
6. In the **Apply to** field, select a class to which the rule applies.
7. In the **Add to ruleset** field, select a ruleset for the rule.
8. **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.
9. Click **Create and open**.
10. 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 a default urgency of 10. The system adds the value in this field to arrive at the initial urgency. **For example:** If you enter 10, the initial urgency is 20.

- 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 DateTime 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.
- 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.
- 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.
- Click **Save**.

What to do next: To facilitate timely resolution of cases and notify case workers, define escalation actions. For more information, see *Defining an escalation action for an unresolved case*.

- [Completing work on time](#)
- [Pega-ProCom agents](#)
- [Processes landing page](#)
- [Creating unit test suites with AUT](#)
- [AUT test suite Create or Save as form](#)
- [More about Skin rules](#)

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.

- 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, and then click **Goal & deadline**.
- 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**.
- If you select to notify participants, click **Manage participants**, select the case participants that you want to notify, and then click **Done**.
- 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**.
- In the **Deadline** section, create a notification by repeating steps 3 through 5.
- Click **Save**.

- [Defining an escalation action for a stage and a process](#)
- [Setting service-level agreements \(SLAs\) for case resolution](#)

Defining an escalation action for a stage and a process

Ensure that your business 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.

Before you begin: Define notification channels. For more information, see [Notification channels](#).

- In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
- 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 elapses, click the stage.
 - To send a notification after a goal or deadline for a process elapses, click the process.
- In the **Properties** pane, click the **Goal & deadline** tab.
- From the **Use Service-level agreement (SLA)** list, select **Custom SLA**.
- In the **Goal** section, in the **Notify** subsection, select the recipients of the notification:
 - To notify the creator of the case, select **Creator**.
 - To notify the participants of the case, select **Participants**.
- If you choose to notify participants, click **Manage participants**, select the case participants that you want to notify, and then click **Done**.
- 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**.
- In the **Deadline** section, create a notification by repeating steps 5 through 7.
- Click **Save**.

What to do next: Notify users that work on mobile devices about elapsed goals and deadlines by enabling push notifications. For more information, see [Sending push notifications about case events](#).

- [Setting service-level agreements \(SLAs\) for stages, processes, and steps](#)
- [Defining an escalation action for a case](#)

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.

- In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
- In the upper-right corner of the **Workflow** tab, click **Life cycle**.
- 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](#).
- In the step property panel, click **Goal & deadline**.
- 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**.
- In the **Deadline** section, define an escalation action by repeating step 5.
- Click **Save**.

- [Assignment shapes in processes](#)
- [Collecting information from a user](#)

Managing work across your team

Achieve the best results while processing your Microjourney by efficiently managing work across your team. When you create a team for users and define routing conditions to assign tasks, you ensure that each task in the case is completed by the team member that is best-suited to the job. As a result, you meet your business goals faster, and to greater effect.

To ensure that you have the right type of resources to resolve your business processes, create a team, and then decide on roles in your team. For example, you can assign one team member as a manager, to monitor and approve the work of the other team members.

When your team is up and running, you can start processing cases by assigning tasks to your team members. To save time and ensure that your application routes tasks to the most appropriate person, you can automate the process of assigning work by configuring custom routing logic and choosing assignees at run time. You can then define the routing conditions based on workload and skills, to ensure that the right team member, with sufficient capacity, receives the task.

- [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.

- [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.

- [Configuring intelligent email routing](#)

To save time and ensure faster resolution of cases, configure your Pega Email Bot™ to react to user requests and automatically respond to emails. By creating conditions for intelligent email routing, the system can then automatically convert email content to a new interaction case for a user, add the email to a work queue, create a top-level case in an application, or send an automatic reply.

- [Switching between applications in Dev Studio](#)

When you work on multiple applications, you can switch between them by using the Application menu.

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.

- [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.

- [Adding a secondary manager to a team](#)

Increase the visibility of your work by associating your team with more than one manager.

- [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.

- [Deleting a team](#)

Delete a team when it does have any associated users or work queues to reduce complexity in your application.

- [Inviting collaborators to your application](#)

Enhance your application and begin processing your business cases by inviting collaborators with different skills and roles.

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 App Studio, navigate to a portal that contains the **My Teams** widget.
2. In the navigation pane, click **My Teams**.
3. In the header of the **My Teams** section, click **Create team**.
4. 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**.
5. Add team members:
 - a. In the **My Teams** section, click the team name.
 - b. 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**.

What to do next: Facilitate collaboration in your team by adding the Pulse gadget to your application. For more information, see [Collaborating with users by using Pulse](#).

- [Creating a work queue](#)

A work queue is a container that holds work that is waiting to be assigned to operators and robotic processes. Create a work queue so that work can be categorized prior to being assigned.

- [Inviting collaborators to your application](#)
- [Removing a team member](#)

Creating a work queue

A work queue is a container that holds work that is waiting to be assigned to operators and robotic processes. Create a work queue so that work can be categorized prior to being assigned.

1. In the header of Dev Studio, click **Create Organization Work queue**.
2. On the **Create Work queue** tab, in the **Short description** field, describe the purpose of the work queue. **For example:** Enter *Loan requests*.
3. In the **Work queue** field, enter a name that clearly conveys the purpose of the work queue and its contents to users of the application.

Note: For a better distinction between work queues and other organization elements, append the wq characters to the work queue name. **Note:** You cannot use pyBulkProcessing as a work queue name. This value is reserved for background bulk processing.

For example: Enter *LoanRequestsWQ*.
4. Click **Create and open** to open the rule form.

Note: The ruleset name assists with application packaging and is not used for assigning work. To change the ruleset, click **Edit** at the top of the form next to the default ruleset name, and enter a new ruleset name or leave the field blank.
5. Specify an organization, division, and unit. For more information, see *Associating a work queue with an organization*.
6. Click **Save**.

- [Creating unit test suites with AUT](#)
- [Defining operator work groups, work queues, and schedules](#)
- [Copying a rule or data instance](#)
- [Creating a specialized or circumstance rule](#)

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**.

Result: The secondary manager can access the team's work queue but cannot complete assignments or receive requests for approval.

- [Creating a team](#)

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. As the application author or case manager, click **My Teams**.
2. In the **Members** section, click the avatar of a team member.
3. Click **Actions Edit profile**.
4. Click **New skill**.
5. 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.
6. 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.
7. Click **Submit**.

- [Creating a team](#)

Deleting a team

Delete a team when it does have any associated users or work queues to reduce complexity in your application.

1. As the application author or case manager, click **My Teams**.
2. Click a team name.
3. On the **Team** page, click **Actions Delete team**.
4. In the **Delete Team** dialog box, enter the reason why the team is no longer required.
5. Click **Submit**.

- [Removing a team member](#)
- [Creating a team](#)

Inviting collaborators to your application

Enhance your application and begin processing your business cases by inviting collaborators with different skills and roles.

For example, you can invite members of a development team so that the team can start working on building your application. When you invite collaborators, you associate them with personas that you create in your application. Personas define type of access that you grant to the users of your application.

Before you begin: Define types of access for the users of your application. See [Granting personas access to channels and pages](#).

1. In the navigation pane of App Studio, click **Users**, and then click **People**.
2. In the working area, click **Invite people to your application**.
3. In the **Add users to application** window, in the autocomplete field, press the Down arrow key, and then select a user name or email address. **Result:** If you enter an email address that does not exist, the system creates a new user for you.
4. In the list, define a type of access that the user has to your application by selecting a role.
When you select a role, you associate a user with a persona. For more information about personas, see [Creating personas](#).
5. Click **Add**.
6. Send an email with an invitation to the user by clicking **Send invitation**.
7. Close the **Add users to application** dialog box.

What to do next: Analyze what data your case requires to reach a resolution by defining data objects. See [Adding data objects to organize data](#).

- [Creating a team](#)

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.

Before you begin: Create an assignment that you want to route. For more information, see [Adding a step to a process](#).

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, select a task that you want to route.
4. In the properties panel, in the **Step** section, 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 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](#).

5. Click **Save**.

- [Assigning users automatically at run time](#)

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 or teams, you can provide conditions that your application evaluates at run time to assign a task.

- [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.

- [Defining the case life cycle](#)
- [Setting service-level agreements \(SLAs\) for stages, processes, and steps](#)
- [Adding optional actions to cases](#)

Assigning users automatically at run time

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 or teams, you can provide conditions that your application evaluates at run time to assign a task.

For example, your application can evaluate which CSR has the most appropriate skill set or workload to complete a step or task.

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 in a process.
4. In the **Step** section of the properties panel, click the **General** tab.
5. In the **Route to** list, select **Use business logic**, and then click the **Configure business logic** icon.
6. Select a routing method:

Actions	Steps
Route the assignment to a specific user	a. In the Route to list, select Operator . b. In the Value field, press the Down arrow key, and then select the name of a user.
Route the assignment to a work queue	a. In the Route to list, select Work queue . b. In the Value field, press the Down arrow key, and then select the name of a work queue.
Route the assignment by using custom business logic	a. In the Route to list, select an option from the Custom section. b. Specify routing conditions by clicking Edit parameters and completing the Edit parameters window. You can route the assignment based on availability and skill set, or based on availability, skill set, and workload.

7. Define the routing condition for the assignment:
 - a. In the list of fields, select the name of a field in your data model.
 - 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. **For example:** Case status is equal to New
 - d. **Optional:** To define more conditions that apply to this routing method, click the **Add row** icon.
8. **Optional:** To support complex business logic, click **Add condition**, and then repeat steps 6 and 7. **Result:** At run time, your application selects the first routing method that meets the conditions that you define.
9. In the **otherwise** section, define a routing method that you want to use when no conditions return a true value:

Actions	Steps
Route the assignment to a specific user	a. In the Route to list, select Operator . b. In the Value field, press the Down arrow key, and then select the name of a user.
Route the assignment to a work queue	a. In the Route to list, select Work queue . b. In the Value field, press the Down arrow key, and then select the name of a work queue.
Route the assignment by using custom business logic	a. In the Route to list, select an option from the Custom section. b. Specify routing conditions by clicking Edit parameters and completing the Edit parameters window. You can route the assignment based on availability and skill set, or based on availability, skill set, and workload.

10. Click **Submit**.

Result: At run time, an application evaluates the conditions and automatically routes an assignment to an appropriate 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](#)

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](#).

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. Choose a type of destination for the assignment:
 - To route the assignment to a specific user or worklist, in the **Assignment type** list, select **Worklist**.
 - To route the assignment to a team or work queue, in the **Assignment type** list, 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. **For example:** You can provide the name of a decision tree that finds a worklist based on the skill set of a user, when you use the *Work-.ToDecisionTree* router activity. **For example:** You can provide the name of a work queue and a list of required skills when you use the *Work-.ToSkilledWorkbasket* router activity.
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](#)

Configuring intelligent email routing

To save time and ensure faster resolution of cases, configure your Pega Email Bot™ to react to user requests and automatically respond to emails. By creating conditions for intelligent email routing, the system can then automatically convert email content to a new interaction case for a user, add the email to a work queue, create a top-level case in an application, or send an automatic reply.

For example, when you configure intelligent email routing to detect a credit card issue in the email subject and a customer email about a problem with their credit card transaction arrives, the email bot automatically routes the email to the correct operator and creates a top-level case to help track the issue in the system.

Before you begin: Define Email channel behavior for the email bot. For more information, see [Defining Email channel behavior](#). Use the customizable Interaction API for text analytics with your email bot to set up routing conditions based on a decisioning result, to determine the outcome of the interaction strategy. The *pyInteractionDF* data flow for the Interaction API performs natural language processing through the iNLP text analyzer for the email bot and feeds the outcome to the interaction strategy. The system captures the strategy outcome in the *pyDecisionResult* property for the *Data-pxStrategyResult* class. For more information, see [Customizable Interaction API for text analytics](#).

1. In the navigation pane of App Studio, click **Channels**.
2. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
3. In the Email channel, click the **Behavior** tab.
4. In the **Intelligent routing** section, in the **Process** list, select a processing mode:
 - To trigger only the first action in the list that matches the conditions, select **First matching action**.

All other actions in the list with matching conditions are not triggered.

- To trigger every *create case* and *automatic reply* action in the list that matches the conditions, and skip all *route to* actions, select **All matching actions**.

If no *create case* or *automatic reply* action in the list match the conditions, the system triggers only the first *route to* action that matches the conditions.

Selecting the routing processing mode enables the system to trigger only the first action or every *create case* and *automatic reply* action in the list with matching

conditions.

5. In the **Intelligent routing** section, click **Add condition**.

6. In the **Action** list, select an action that you want to run automatically for email routing. **For example: Create case**

7. In the **Value** field, enter an email template, case name, operator, or work queue. **For example:** For the **Create case** action, enter a case type name: Insurance Claim

8. Define the validation conditions for a field on the form as the criteria for carrying out the routing action:

Choices	Actions
Detect email header content	a. In the When list, select Header and the name of the email field that you want to validate, for example: Subject. b. In the comparator list, select the test that you want to perform on the field, for example: Contains. c. In the value field, enter or select a value to compare against the user input, for example: price.
Detect an entity - a word or phrase in an email	a. In the When list, select NLP Entity , and an entity name, for example: #Email. b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value field, enter a value to compare against the user input, for example: Bob@example.com.
Detect the topic - the subject matter of an email	a. In the When list, select NLP Topic . b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value field, enter a value to compare against the user input, for example: Insurance Quote.
Detect a decisioning result - the outcome of the interaction strategy	a. In the When list, select NLP Decisioning result . b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value field, enter a value to compare against the user input. The email bot can check whether the decisioning result is empty, is not empty, or contains a specified value as the outcome of the interaction strategy, for example, a postal code, or a company or product name. For more information, see Customizable Interaction API for text analytics.
Detect the positive, negative, or neutral sentiment of an email	a. In the When list, select NLP Sentiment . b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value list, select what sentiment to compare against the user input, for example: Positive.
Detect the language of an email	a. In the When list, select NLP Language . b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value list, select a language to compare against the user input, for example: French.
Use a when rule	a. In the When list, select Condition. b. In the field to the right of the When list, enter or select a when rule that you want to use as the condition for carrying out the action. For example, you can configure a routing condition to route the email to an operator <i>Andrew Sullivan</i> when the <i>pyIsInWorkList</i> rule is true.

Each routing condition specifies a single way that the email bot automatically responds to an email.

9. **Optional:** To add more automatic responses based on email header content, NLP text analysis of the email, or a when rule, repeat step 5 through 8 for other routing conditions.

10. Specify how your email bot responds when you do not define any criteria or when the email content does not meet any of the criteria that you specified in step 8 by defining the default action:

a. In the **Otherwise** area, in the **Action** list, click an action to perform.

b. In the **Value** field, enter an email template, case name, operator name, or work queue.

11. Click **Save**. **For example:**

Intelligent routing configuration for a first matching action

Intelligent routing

Process: First matching action

Triggers only the first action that matches conditions.

Action

Create case

Value

Address Update

When

Sentiment

Is equal

Negative

Action

Route to work queue

Value

NewApp:Users

When

Topic

Is equal

+ Add condition

Otherwise

Action

Route to operator

Value

Intelligent routing

Process: First matching action

Triggers only the first action that matches conditions.

Action	Value
Create case	Address Update

When Sentiment Is equal Negative +

Action	Value
Route to work queue	NewApp:Users

When Topic Is equal +

+ Add condition

Otherwise

Action	Value
Route to operator	

12. Verify that the intelligent email routing works as expected.
For more information, see [Testing email routing conditions](#).

What to do next: Train the model for the email bot by using sample data. For more information, see [Training data for the Email channel](#).

- [Testing email routing conditions](#)

When you configure an Email channel, define routing conditions to improve the responses that Pega Email Bot™ provides to an email that you receive. Before training and moving the email bot natural language processing (NLP) model to a production environment, you can use the test console to verify all the routing conditions for intelligent email routing.

- [Implementing an email bot for an Email channel](#)
- [Pega Email Bot overview](#)

Testing email routing conditions

When you configure an Email channel, define routing conditions to improve the responses that Pega Email Bot™ provides to an email that you receive. Before training and moving the email bot natural language processing (NLP) model to a production environment, you can use the test console to verify all the routing conditions for intelligent email routing.

Before you begin: Define routing conditions for the email bot based on email header content, NLP text analysis, or a when rule. For more information, see [Configuring intelligent email routing](#).

You can test whether your email bot triggers the correct routing actions that you define for each routing condition case. In addition, you can verify sentiment detection.

1. In the navigation pane of App Studio, click **Channels**.
2. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
3. In the Email channel, click the **Behavior** tab.
4. Click **Test**.
5. In the **Test routing conditions** window, verify a routing condition:
 - a. Challenge criteria for a routing condition by entering the parameters of a sample email.
 - b. Click **Test**.
 - c. In the **Outcomes** section, verify that the correct action occurs for the routing condition.
 - d. Click **Close**.
6. **Optional:** To test all routing conditions, repeat step 5 for each routing condition.

What to do next: After verifying that each of the routing actions occurs when the criteria hold true, train the model for the email bot by using sample data. For more information, see [Training data for the Email channel](#).

- [Configuring intelligent email routing](#)
- [Defining Email channel behavior](#)

Switching between applications in Dev Studio

When you work on multiple applications, you can switch between them by using the **Application** menu.

1. In Dev Studio, click the name of your current application to access the **Application** menu.
2. Click **Switch Application**.
3. From the alphabetized list, select an application to become your current application.
Note: The list of available applications is based on the access groups defined on your Operator ID form. If more than one access group points to the same application, the **Switch Application** menu displays the application label followed by the access group in parenthesis.

- [Adding the My Applications gadget](#)

To conveniently navigate between your applications, add the My Applications gadget to your existing portal or application by embedding a header and a menu.

- [Setting the default landing page to My Applications](#)

To navigate conveniently between your applications, access the My Applications gadget after you log in to your portal. You can use the gadget to see an overview of all your applications with short descriptions, and then search for and select an application that you want to open.

Adding the My Applications gadget

To conveniently navigate between your applications, add the *My Applications* gadget to your existing portal or application by embedding a header and a menu.

By using the *My Applications* gadget, you can see an overview of your applications with short descriptions, customize the icons of your applications, and search the applications by name. By default, the *My Applications* gadget is available in App Studio.

Before you begin: Ensure that your application uses UI Kit 14 or any of its later versions.

1. In Dev Studio, open an application in which you want to add the **My Applications** gadget.
For more information, see [Switching between applications in Dev Studio](#).
2. In the navigation panel, click **Records User Interface Section** , and then open the **pyPortalHeader** section.
3. Right-click the embedded **pyAppLauncher** section, and then click **Copy**.
4. Open the section where you want to add the **My Applications** gadget.
5. Right-click the section, and then click **Paste**.
6. Open the **pyPortalHeader** section.
7. Repeat steps [3](#) through [5](#) for the **My Applications** icon.
8. Click **Save**.

- [Presentation Tab](#)

Setting the default landing page to My Applications

To navigate conveniently between your applications, access the *My Applications* gadget after you log in to your portal. You can use the gadget to see an overview of all your applications with short descriptions, and then search for and select an application that you want to open.

By default, the **My Applications** gadget is available in App Studio.

Before you begin: To switch between applications in portals other than App Studio, add the gadget. For more information, see [Adding the My Applications gadget](#).

1. In Dev Studio, open an application in which you want to set the **My Applications** gadget as default.
For more information, see [Switching between applications in Dev Studio](#).
2. Hover over your operator icon, and then click **Access group**.
3. In the **Available portals** section, click **Add portal**.
4. In the new **Name** field, enter pyApplications.
5. In the **Default** column, select the radio button for your new portal.
6. Click **Save**.

- [Building your first application](#)

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.

- [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.

- [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.

- [Notification channels](#)

A notification channel is a carrier that the application uses to notify users about events at run time. You can create notifications for one or all of the available notification channels - web gadget, email, and mobile.

- [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.

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**.

- [Creating unit test suites with AUT](#)
- [Copying a rule or data instance](#)
- [Creating a specialized or circumstance rule](#)

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 unit test suites with AUT](#)
- [Creating a notification](#)

- [Copying a rule or data instance](#)
- [Creating a specialized or circumstance rule](#)

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**.

- [Creating unit test suites with AUT](#)
- [Creating a notification](#)
- [Copying a rule or data instance](#)
- [Creating a specialized or circumstance rule](#)

Notification channels

A notification channel is a carrier that the application uses to notify users about events at run time. You can create notifications for one or all of the available notification channels – web gadget, email, and mobile.

You can configure notification channels in Dev Studio:

- [Configure a web gadget notification channel](#)
- [Configure an email notification channel](#)
- [Configure a mobile notification channel](#)

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 the *Adding a custom notification channel* article on Pega Community. After creating the custom channel, you can configure it on the **Channels** tab for your notification.

- [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.

- [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.

- [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.

- [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.

- [Creating unit test suites with AUT](#)
- [Creating a notification](#)
- [Configuring a web gadget notification channel](#)
- [Configuring an email notification channel](#)
- [Configuring a mobile notification channel](#)
- [Configuring a web gadget notification channel](#)

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](#)
- [Notification channels](#)
- [Creating unit test suites with AUT](#)
- [Creating a notification](#)

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.

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](#).
 - To disable email notifications for all users, on the **Channels** tab, in the **Email** section, clear the **Enable email** check box, and then go to [6](#).
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](#)
 - [Notification channels](#)
 - [Creating unit test suites with AUT](#)
 - [Creating a notification](#)

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](#)
- [Notification channels](#)
- [Creating unit test suites with AUT](#)
- [Creating a notification](#)

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](#)
- [Creating unit test suites with AUT](#)
- [Creating a notification](#)

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.

What to do next: Configure Pulse, relevant case types, and mobile settings to support the notification preferences that you choose.

- [Notifying participants about events](#)
- [Engaging mobile users with push notifications](#)
- [Case processing](#)
- [Creating unit test suites with AUT](#)

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.

- [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.

- [Creating ad hoc cases](#)

You can handle business exceptions or track related tasks in the case that you currently process by creating an ad hoc case. With 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.

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.

- [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.

- [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.

- [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.

- [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.

- [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.

- [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.

- [Automating work by creating case types](#)

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.

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 **Optional actions**.
3. In the **Case wide actions** section, click **Action**.
4. 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 a step to a process](#).

- 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**.
5. In the text field, replace the default label for the action with a descriptive action name.
6. Click **Save**.

Result:

At run time, optional tasks appear in the header of a case.

- [Defining the case life cycle](#)
- [Adding optional actions to stages](#)
- [Displaying optional actions conditionally](#)

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.

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 **Optional actions**.
3. In the **Stage-only actions** section, under the stage that you want to supplement with an optional action, click **Action**.
4. 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 a step to a process](#).

- 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**.
5. In the text field, replace the default label for the action with a descriptive action name.
 6. Click **Save**.

Result:

At run time, the action appears in the header of a case when the case enters the stage.

- [Defining the case life cycle](#)
- [Adding optional actions to cases](#)

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.

Before you begin: Add an optional action to your case life cycle. For more information, see [Adding optional actions to cases](#) and [Adding optional actions to stages](#).

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**.
3. 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.
4. In the **User action** section, define the conditions for displaying the task:

Choices	Actions
Always display the task	In the Visible list, select Always .
Create a custom condition for displaying the task	a. In the Visible list, select Custom condition. b. Next to the list, click the Configure condition icon. c. In the Configure condition window, select a condition, a comparator, and a value to compare with the condition. For example: Case status is equal to New d. Optional: To define more conditions, click the Add row icon, and then repeat substep 4.c. e. Optional: To group the conditions, select comparators from the list. For example: To make only one condition compulsory, in the list of comparators, select OR. f. Optional: To reuse the condition for other steps in the future, click Actions Add to when conditions library. g. Click Submit.
Use an existing condition for displaying the task	a. In the Visible list, select Existing condition. b. In the list of conditions, select a value.

5. Click **Save**.

Result: 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](#)

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 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**.
3. In the **Case wide actions** section, click **Action**.
4. 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**.
5. **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.
6. Click **Save**.

Result: The supporting process runs as a step in the current stage of the case. **What to do next:** Provide relevant tasks for your process and decide when a user can work on the process:

- Add meaningful actions to new processes or supplement existing processes with more tasks by adding steps. For more information, see [Adding single steps to processes](#).
- Define when the user sees the process by configuring visibility conditions. For more information, see [Displaying supporting processes conditionally](#).
- [Adding supporting processes to stages](#)
- [Displaying supporting processes conditionally](#)

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 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**.
3. In the **Stage-only actions** section, click **Action** below the stage to which you want to add a supporting process.
4. Choose a supporting process:
 - To add a new process, click **Process**.
 - To add an existing process, click **More Processes**, and then select a process.
5. Replace the default label for the process with a name that describes the purpose.
6. Click **Save**.

Result: At run time, you can refer to the header of a case to find the supporting processes that are attached to the current stage. **What to do next:** Provide relevant tasks for a supporting process and define when a user can work on the process:

- Add meaningful actions to new processes, or supplement existing processes with more tasks, by adding steps. For more information, see [Adding single steps to processes](#).
- Define the conditions for when users see the process. For more information, see [Displaying supporting processes conditionally](#).
- [Adding a process to a stage](#)
- [Adding optional actions to stages](#)
- [Adding a step to a process](#)

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 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**.
3. 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.
4. In the **Process** properties panel, define the conditions for displaying the supporting process:

Choices	Actions
Always display the process	In the Visible list, select Always .
Create a custom condition for displaying the process	a. In the Visible list, select Custom condition. b. Next to the list, click the Configure condition icon. c. In the Configure condition window, select a condition, a comparator, and a value to compare with the condition. For example: Case status is equal to New d. Optional: To define more conditions, click the Add row icon, and then repeat substep 4.c. e. Optional: To group the conditions, select comparators from the list. For example: To make only one condition compulsory, from the list, select OR. f. Optional: To reuse the condition in the future, click Actions Add to when conditions library. g. Click Submit.
Use an existing condition for displaying the process	a. In the Visible list, select Existing condition. b. In the list of conditions, select a value.

5. Click **Save**.

Result: At run time, when a case reaches a supporting process that meets the conditions, the user sees the process.

- [Adding supporting processes to cases](#)
- [Adding supporting processes to stages](#)

Creating ad hoc cases

You can handle business exceptions or track related tasks in the case that you currently process by creating an ad hoc case. With 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.

Before you begin: Ensure that your dashboard includes the My Cases widget. For more information, see [Adding a widget to your dashboard](#).

1. In the Case Manager portal, in the **My Cases** widget, in the **Create an ad hoc case** text field, enter a name for your ad hoc case.
2. Click **Quick create**.
3. Add tasks to the ad hoc case:
 - a. In the **My cases** widget, click the name of the ad hoc case that you created.
 - b. In the upper-right corner of the portal, click **Actions 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.
 - d. Click **Submit**.

Result: Your application displays the task. Users can make notes in the tasks when they complete them.

Result: If you create multiple tasks in an ad hoc case, your application displays the next task after the user resolves the current task.

- [Configuring and working with optional actions in case types](#)

Enriching cases with external data

Expand your Microjourney to take advantage of functionalities outside of your application, by connecting cases to external data sources. By enriching cases with external data, you can precisely cater to the needs of your customers, and resolve business cases more efficiently.

For example, you can increase the functionality of your application by assigning tasks to an external application, or to a robotic queue. When you reuse external resources that add value to your application, you speed up your Microjourney and meet your business goals in a more efficient way.

- [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.

- [Configuring external storage options for attachments](#)

Use external storage to attach content to a case or Pulse conversation.

- [Storing case attachments using external storage](#)

You can configure your application to store case and Pulse attachments outside the Pega Platform database. By using external resources, you can upload large files without causing out-of-memory errors.

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:

- [Assigning a task to a robotic work queue](#)

Use the Assign to Robot Queue shape in a flow to integrate Robotic Process Automation (RPA) with a case. By defining the tasks that a virtual machine (VM) can run, you can reduce the need for human input in your application and resolve cases faster.

- [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.

- [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.

- [Running a decision strategy from a flow](#)

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 flows, you create decision logic that is contextual for each case.

- [Calling an activity or an automation from a flow](#)

You can use the Utility shape to call an activity or an automation from a flow. By using an activity or automation, you can use fewer shapes in your flow and perform processing that does not require human input.

- [Creating cases from a web mashup](#)

To manage business processes of customers who use external software, embed a case into an external application or a web page by using a web mashup. You can generate an HTML code that you insert into an external web page or application, so that users can create cases even without access to Pega Platform software.

- [Creating a stand-alone process](#)
- [Calling one process from another process](#)

Assigning a task to a robotic work queue

Use the **Assign to Robot Queue** shape in a flow to integrate Robotic Process Automation (RPA) with a case. By defining the tasks that a virtual machine (VM) can run, you can reduce the need for human input in your application and resolve cases faster.

1. Add the **Assign to Robot Queue** shape to your flow:
 - a. Open a flow by searching for it or by using the Application Explorer. For more information, see [Finding rules by class](#).
 - b. On the **Diagram** tab of the Flow form, click the **Flow shapes** icon, and select **Advanced Shapes Assign to Robot Queue**.
 - c. Drag the **Assign to Robot Queue** shape to a position on the flow diagram, based on the order of events in the flow.
 - d. Connect the **Assign to Robot Queue** shape, by dragging connector end points to connection points on different shapes in the flow.
2. Open the **Robot queue** dialog box by double-clicking the **Assign to Robot Queue** shape. Robot queue</wintitle> dialog box.">
3. Define the context of the robotic automation, which runs as a subprocess of the current flow:
 - To work on the current case, from the **Define flow** list, select **On current page**.
 - To work on a specific case, from the **Define flow** list, select **On specific work item**. Complete the **Work property**, **Class**, and **Page name** fields. **Note:**

If the case is open as a page on the clipboard, you can leave the **Work property** field blank.

If the case does not already have a page and you do not provide a page name, your application creates a page named pyNextObj, pyNextObj_1, and so on.
 - To work on an embedded page, from the **Define flow** list, select **On embedded page**. Complete the **Page property** and **Class field** names. **Note:**

The **Page property** field must use a mode of Page, Page List, or Page Group.

If you want to use a class that is not known at design time, in the **Class** field, you must enter a property reference or expression.
4. In the **Route to queue** field, enter the name of the work queue from which the VM retrieves assignments. **Result:** The Route to robot queue shape creates the assignment that the VM retrieves.
5. In the **Run robotic automation** field, enter the automation that the virtual machine runs after retrieving the assignment.
6. **Optional:** To update the status of a case by synchronizing it with the state of the robotic assignment, set the case status changes in the **Set case status when** section. For each of the assignment states that you want to map to a case status, perform one of the following actions:
 - To select a case status that is triggered by a robotic assignment state, press the Down Arrow key in the corresponding field.
 - To create a custom status, enter a unique name in the field and click on the **Open rule** icon.

Note:

If you do not select a status in a corresponding field, the previous state of the case will be carried forward.

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
Assignment is queued	Change case status when a robotic assignment is added to a work queue. Note: Work queues are called <i>assignment types</i> in Pega Robot Manager.
Robotic Automation is in progress	Change case status when a robot pulls the assignment and starts working on it.
Robotic Automation has conflicting results	Change case status when a robotic assignment is completed but returns data that fails validations specified for the case.
Robotic Automation fails	Change case status when a robotic assignment could not be completed and failed.
Robotic Automation times out	Change case status when a robotic assignment could not be completed in the threshold time specified in the assignment type/work queue.

7. **Optional:** Validate case data after the assignment is processed by pressing the **Down Arrow** key in the **Assignment validation criteria** field and selecting a validation rule.
8. **Optional:** Apply a goal and deadline to the assignment by pressing the **Down Arrow** key in the **Assignment service level** field, and select the name of a service-level agreement.
9. **Optional:** Allow the main flow to continue processing without waiting for the robotic automation to complete by selecting the **Spinoff flow** check box.
10. Close the **Robot queue** dialog box by clicking **Submit**.
11. Click **Save**.

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.

Before you begin: To assign a task to an external application:

- Ensure that the process for which you want to use an Assignment Service shape already exists. For more information, see [Creating a top-level case type](#).
- Build your own connect rule, and then build a connect activity that maps the data from the service that you call into your case. For more information, see [Creating a REST integration](#).

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](#) 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 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 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. **For example:** You can set a service level for the task to begin escalation processing when the application fails to get a response from the external system after 60 seconds (which might indicate that the external system is unavailable or that other technical problems are present). For more information, see [Errors returned by a connector](#) and [Handling connector exceptions](#).
 - 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](#).
 - 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](#).
3. Click **Submit**.
 4. Click **Save**.

- [Setting a service-level agreement for the Assignment Service shape](#)

You can associate a service-level agreement with the Assignment Service shape in a flow. 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.

- [Integrating a case with an external resource](#)
- [Activity form - how to create activities for flows](#)

Setting a service-level agreement for the Assignment Service shape

You can associate a service-level agreement with the Assignment Service shape in a flow. 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.

Service-level agreements on shapes are independent from service-level agreements that are set on a flow or case type.

1. Open a flow by searching for it or by using the Application Explorer.
2. On the Diagram tab, double-click the Assignment Service shape in the flow diagram.
3. In the Service level field, press the Down Arrow key and select the name of a service-level agreement.

If you do not select an option, the *Work-.Default* service-level agreement is used.

4. Click **Submit** to close the property panel.
5. Click **Save**.

Note: Open the Service Level Agreement form for more configuration options, such as escalation actions that you can run when a goal or deadline are past due.

- [Assigning a task to an external application](#)

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.

Before you begin: To connect a flow to an external application:

- Ensure that the process in which you want to use an Integrator shape already exists. For more information, see [Creating a top-level case type](#).
- Build your own Connect rule, and then build a Connect activity that maps the data from the service that you called into your case. For more information, see [Creating a REST integration](#).

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 pages](#).

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](#).
 - 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](#).
3. Click **Submit** to close the property pane.
4. Click **Save**.

- [Creating a stand-alone process](#)
- [Creating unit test suites with AUT](#)

Running a decision strategy from a flow

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 flows, you create decision logic that is contextual for each case.

1. Add the Run Data Flow shape to your flow.
 - a. Open a flow by searching for it or by using the Application Explorer. For more information, see [Finding rules by class](#).
 - 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](#)
 - [Types of data flows](#)
 - [About Strategy rules](#)
 - [Flow shapes](#)

Calling an activity or an automation from a flow

You can use the Utility shape to call an activity or an automation from a flow. By using an activity or automation, you can use fewer shapes in your flow and perform processing that does not require human input.

1. Open a flow by searching for it or by using the Application Explorer. For more information, see [Finding rules by class](#).
2. On the **Diagram** tab, click the **Flow shapes** icon, and then click **Utility**.
3. Drag the Utility shape to a position on the flow diagram, based on the order of events in the flow.
4. Connect the Utility shape, by dragging connector end points to connection points on different shapes in the flow.
5. Double-click the Utility shape to open the property panel.
6. In the **Select type of rule** field, select the type of rule.
7. Complete the following fields based on the type of rule you are using:
 - Activity – In the **Rule** field, select the name of an activity that is set to type Utility. If the activity supports input parameters, additional fields are displayed for you to define name-value pairs.
 - Automation – In the **Resource** field, select type of automation you are using, then in the **Name** field, select the automation to use. If the activity supports input parameters, additional fields are displayed for you to define name-value pairs.
8. Click **Submit** to close the property panel.
9. Click **Save**.
 - [Flow shapes](#)
 - [Integrating a case with an external resource](#)
 - [Creating unit test suites with AUT](#)

Creating cases from a web mashup

To manage business processes of customers who use external software, embed a case into an external application or a web page by using a web mashup. You can generate an HTML code that you insert into an external web page or application, so that users can create cases even without access to Pega Platform software.

You can also test how the mashup code that you generate looks in a mobile app, or preview a channel to verify that mashup code displays correctly on different types of devices.

For example, you can embed a case type mashup gadget for processing job applications on a web page that displays job offers.

Before you begin: Create a case type that you want to insert into an external web page or application. For more information, see [Creating a case type](#).

1. In the navigation pane of App Studio, click **Channels**.
2. In the **Create new channel interface** section, click **Web mashup**.
3. In the **Basic options** section, in the **Name** field, enter a name for the Mashup channel interface that you want to use to create case types.
4. **Optional:** To distinguish this channel interface from other Mashup channel interfaces, in the **Description** field, specify the purpose of this channel interface. **For example:** Enter Case type for processing job applications.
5. In the **URL** field, enter the URL for a Pega Platform instance.
The URL format is `https://host/prweb/app/applicationname_sessionidentifier`.
For example: `http://sample.pega.com:8080/prweb/app/sampleapp_1234`
6. **Optional:** To generate the mashup snippet with URL encryption, turn on the **Use encryption** toggle.
7. In the **Configuration** section, in the **Action** list, select **Create a new case type**.
8. In the **Case type** list, select a case type that you want to use.
9. In the **Flow name** field, select the flow action that starts the case type.
By default, the flow action that starts a case type is *pyStartCase*.
10. In the **Thread name** field, enter the name of the thread that handles the mashup.
11. **Optional:** To load the mashup gadget as the last item on the page, turn on the **Defer mashup load** toggle.
12. Configure output options for the mashup gadget:
 - a. In the **Skin** list, select a skin that you want to use for the case type mashup gadget.
 - b. In the **Iframe resizing** list, select the resize mode for the mashup gadget.
 - c. In the **Initial skeleton** list, select the skeleton to organize content within the mashup gadget.
Tip: You can check how the skeleton organizes content by clicking **Skeleton preview**.
13. Click **Generate mashup code**, and then, in the **Mashup code** window, click **Copy**.
14. Click **Save**.

What to do next: Paste the HTML code into a static web page or webview of an external web application.

- [Defining the case life cycle](#)
- [Configuring views for cases](#)
- [Configuring case creation](#)

Configuring external storage options for attachments

Use external storage to attach content to a case or Pulse conversation.

Before you begin: If you are using a web storage provider, ensure that web storage providers are correctly configured. For more information, see the article *Downloading and configuring pluggable web storage provider components* on Pega Community.

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 & security** tab.
3. In the **Content sourcing** section, configure one or more external storage systems.
 - To use external content for case attachments only:
 1. Select **Source from CMIS systems**.
 2. **Optional:** To change the label that users see when they add an attachment from this system, enter a description in the **Label for this content source** field.

- To use external content for case or Pulse attachments from a web service provider:
 1. Select **Source from web storage providers**.
 2. In the **Provider** list, select an installed web storage provider for which you have an account and a content location.
 3. In the **Authentication profile** field, press the Down Arrow key and then select an authentication profile of type OAuth 2.0. **Note:** You must use the same authentication profile to store content in and source content from the web storage provider.
 4. **Optional:** To change the label that users see when they add an attachment from this web storage provider, enter a description in the **Label** field.
 5. **Optional:** To use content from more than one web storage provider, click **Add** and repeat these steps to configure another provider.
 - To use external content for case or Pulse attachments from a repository:
 1. Select **Source from repositories**.
 2. In the **Repository** list, select a repository for which you have an account and a contents location.
 3. Click **Browse**.
 4. Select the folder from which to source attachments.
 5. Click **Select**.
 6. **Optional:** To change the label that users see when they add an attachment from this web storage provider, enter a description in the **Label** field.
 7. **Optional:** To use content from more than one folder or web storage provider, click **Add**.
4. Click **Save**.

Result: Content from external storage is available when you add an attachment to a case or Pulse conversation, based on the sources that you configure.

- [Adding a case or Pulse attachment from external content locations](#)

When you work on a case, you can add content from external content locations to a case. You can source content from CMIS systems, web storage, or file storage repositories.

- [Adding a case attachment from a CMIS system](#)

When you work on a case, you can add content from a CMIS system to the case in the Attachments section.

- [Managing a case attachment stored in a CMIS system](#)

Manage versions of a case attachment that is stored in a CMIS system at run time..

- [File and content storage](#)
- [Storing case attachments using external storage](#)
- [Creating an authentication profile](#)

Adding a case or Pulse attachment from external content locations

When you work on a case, you can add content from external content locations to a case. You can source content from CMIS systems, web storage, or file storage repositories.

Note: When using database storage, you can attach multiple files with the same name to the same case. For external storage, you cannot attach multiple files with the same name to the same case because storage providers treat the file name plus the path as a key. Attempting to attach more than one file with the same name to a case results in an error. You can, however, attach files with the same name to different cases.

1. Add a case attachment by completing the following steps:
 - a. Click **Attach new** in the **Attachments** section and then click the menu option that allows you to add files from an external system.
This option is the value that you entered in the **Label** field that is displayed on the **Integration & security** tab of your application rule. **Note:** If you are connecting to your external storage account from this application for the first time, you may be prompted to authenticate and authorize the connection.
 - b. Browse to and attach a file.

Result:

The case attachment appears in the **Attachments** section.

Tip: To remove a case attachment, click the **Delete** icon next to the name of the attachment.

2. Add a Pulse attachment by completing the following steps:
 - a. Optional: Enter a message in the field that appears below the **Pulse** section.
 - b. Click **Upload external** in the **Pulse** section and click the menu option that allows you to add files from an external system.
This option is the value that you entered in the **Label** field on the **Integration & security** tab of your application rule.
 - c. Browse to and attach a file from your external system.
 - d. Click **Post**.

Result: The Pulse message and attachment appear in the **Pulse** section. **Tip:** To remove a Pulse attachment, click the **Close** icon next to the Pulse message. The system deletes both the Pulse message and attachment.

- [Configuring external storage options for attachments](#)
- [Creating an authentication profile](#)
- [Attaching content to a case](#)
- [Attachment types](#)

Adding a case attachment from a CMIS system

When you work on a case, you can add content from a CMIS system to the case in the **Attachments** section.

For information about configuring your application to use a CMIS system, see *Managing case attachments with CMIS systems* on the Pega Community.

1. Click **Attach new** in the **Attachments** section and click the value that you entered in the **Label for 'Upload CMIS content' menu item** field on the **Integration & security** tab of the application rule.
2. Browse and attach a file from one or more CMIS systems. These systems correspond to the *Connect-CMIS* records that are available for the current operator.

The case attachment now appears in the **Attachments** section.

Note: To remove a case attachment, click the **Delete** icon next to the name of the attachment.

- [Configuring external storage options for attachments](#)
- [Managing a case attachment stored in a CMIS system](#)
- [Creating unit test suites with AUT](#)
- [Attachment types](#)

Managing a case attachment stored in a CMIS system

Manage versions of a case attachment that is stored in a CMIS system at run time..

1. Click the **Gear** icon next to the name of the attachment.
2. In the **Manage attachment** window, the file attachment details such as the name, category, description, and version history are displayed. Download the current or an earlier version by clicking the file name for the appropriate version in the **Version history** section.
3. **Optional:** To prevent other users from modifying the file attachment, select **Lock this attachment for local editing**
4. Check in a new version of the file by clicking **Upload new version** and optionally add a comment.

Result: The new version now appears in the **Version history** section.

- [Adding a case attachment from a CMIS system](#)
- [Creating unit test suites with AUT](#)
- [Attachment types](#)

Storing case attachments using external storage

You can configure your application to store case and Pulse attachments outside the Pega Platform database. By using external resources, you can upload large files without causing out-of-memory errors.

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 & security** tab.
3. In the **Content storage** section, select and configure a storage type.

For more information about supported storage types, see [File and content storage](#).

- **Store in Pega database** – Pega database is the default option for on-premises deployments.
- **Store in CMIS system:** In the **Connector name** field, press the Down Arrow key, and then select a Connect CMIS rule that defines connections to CMIS repositories.
- **Store in web storage provider** **Note:** Ensure that web storage providers are correctly configured. For more information, see [Installing components](#).
 1. In the **Provider** list, select an installed storage provider.
 2. In the **Authentication profile** field, press the Down Arrow key, and then select an authentication profile of type OAuth 2.0 that authenticates the connection to the provider that you use. **Note:** You must use the same authentication profile for both storing or sourcing content from a web storage provider.
 3. Click **Browse**, and then select the folder that stores the case and Pulse attachments.
 4. If you are connecting to your web storage provider account from this application for the first time, follow the prompts to authenticate and authorize the connection with your web storage provider.
- **Store in repository**
 1. In the **Repository** list, select a data repository.
 2. Click **Browse**, and then select the folder that stores the case and Pulse attachments.
- 4. **Optional:** To prompt users to decide whether an attachment is removed from the external content management system when they remove it from a case, select the **Prompt users for confirmation before deleting content** check box.
- 5. Click **Save**.

Result: New attachments that you create are stored in the location that you choose.

- [Requirements and restrictions for case attachments in a file storage repository](#)

You can configure your application to store case attachments in a repository. When you use a repository, Pega Platform appends the case ID and time stamp to the attachment name. For example, an attachment called loanapplication.pdf becomes loanapplication_19691231T115959999_CASE-1234.pdf.

- [Migrating case attachments to a repository](#)

Reduce the size of your Pega database and improve performance by migrating your application's case attachments from Pega database storage to a repository.

- [Configuring external storage options for attachments](#)
- [Managing a case attachment stored in a CMIS system](#)
- [Creating an authentication profile](#)
- [AUT test suite Create or Save as form](#)
- [Creating a repository](#)
- [Requirements and restrictions for case attachments in a file storage repository](#)

Requirements and restrictions for case attachments in a file storage repository

You can configure your application to store case attachments in a repository. When you use a repository, Pega Platform appends the case ID and time stamp to the attachment name. For example, an attachment called loanapplication.pdf becomes loanapplication_19691231T115959999_CASE-1234.pdf.

Consider the following requirements and restrictions to ensure that your attachments are stored and retrieved correctly from a repository.

Requirements

- You must use the *D_pxGetFile* API to access the contents of an attachment.
- Your email listener service package must enforce basic authentication.
- Your email listener must be configured with an operator whose primary application uses the repository that your application uses.

Restrictions

- Repositories support file attachments only. All other types of attachments are stored in a database.
- You cannot change the content storage type from repository to a different type.
- Once you switch from database storage to repository storage, you cannot switch back to database storage without losing access to the attachments stored 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 are always stored in a database.
- Because the application Pulse gadget cannot prompt for authentication details, Pulse attachments that you provide on the **Application Overview** landing page are stored in the repository only when your session is already authenticated or when the repository supports non-interactive authentication. If neither of these conditions is true, then an error occurs and the operation fails.
- The following features are not supported for custom repositories during DevOps:
 - Import/export
 - File listener
- For large images and .pdf attachments, poster creation is skipped and creates a placeholder thumbnail.

Multiple files with the same name - differences between database storage and repository storage

When using database storage, you can attach multiple files with the same name to the same case. For external storage, you cannot attach multiple files with the same name to the same case because storage providers treat the file name plus the path as a key. Attempting to attach more than one file with the same name to a case results in an error. You can, however; attach files with the same name to different cases.

- [Storing case attachments using external storage](#)
- [Creating a repository](#)
- [Configuring email listener routing](#)
- [Creating unit test suites with AUT](#)
- [Attachment types](#)

Migrating case attachments to a repository

Reduce the size of your Pega database and improve performance by migrating your application's case attachments from Pega database storage to a repository.

Before you begin: Ensure that the *pzAttachmentMigrationProcessor* queue processor is running. In Admin Studio, click **Resources Queue processors** , and then click **Start** in the **State** field for *pzAttachmentMigrationProcessor*.

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 click **Select**.
4. Click **Start Migration**.
5. Click **Refresh** to update the display.

Result: When the migration is finished, the results include how many files migrated. For the files that were not migrated, the work object ID is displayed along with a description of the error that you can use for troubleshooting. **What to do next:** Ensure that the repository that the files were migrated to is the same as the repository being used for new attachment storage. For more information, see [Configuring external storage options for attachments](#).

- [Storing case attachments using external storage](#)

Designing surveys

Gather information quickly from users while they process a case by designing a survey. When users complete a survey, 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 surveys, 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 surveys. For instance, you can create a question page that captures details about customer satisfaction, and then reuse that question page in surveys about different business processes.

- [Best practices for creating a survey](#)

You can use surveys to gather information from users. Before you create a survey, review these best practices.

- [Creating a survey](#)

Create a survey to provide a structure for the questions that users answer in a case.

- [Fine-tuning your survey](#)

Fine-tune the way that your survey displays questions and evaluates answers to ensure that you collect valid information in a case.

- [Running a survey in a case](#)

Best practices for creating a survey

You can use surveys to gather information from users. Before you create a survey, review these best practices.

The following tips can help you create surveys that are easy to use and maintain:

- Group questions in a logical way so that users stay focused and can provide answers quickly.
- Use question pages where possible because they are reusable.
- Limit the total number of questions in the survey because the quality of answers tends to decline as the length of a survey increases.

You can also split up a survey into smaller surveys that are called in succession.

- Limit the number of questions on each question page because many questions can lead to scrolling at run time.
- Use concise questions.

- Ensure that the format of the question matches the type of question that you are asking.

For example, a question that asks users to rate their customer satisfaction cannot be answered with a single check box.

- Proofread your survey for grammatical errors, duplicate questions, and any acronyms or terms that are not explained.
- Test your survey by asking potential users for feedback on clarity, length, and ease of use.

- [Designing surveys](#)

- [Requirements for surveys in offline mode](#)

Creating a survey

Create a survey to provide a structure for the questions that users answer in a case.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click **Create survey**.
3. In the **Create survey** dialog box, enter the name of your survey, and then click **Submit**.

- [Adding a question page to a survey](#)

Add a question page to a survey to organize related questions.

- [Adding a question to a question page](#)

Add a question to a question page to prompt users for information in a format that you define.

- [Creating a stand-alone question](#)

Create a stand-alone question, which you can reference outside the context of a survey, to support information gathering when a case is offline.

- [Configuring advanced options for a question page](#)

Adjust question pages to meet your specific business needs by configuring advanced options. For example, categorize question pages to efficiently organize future reports, or to add an activity that performs an action before displaying the question page.

- [Reusing answers in survey questions](#)

To create more personalized and comprehensive surveys, reuse answers to questions in the subsequent questions. For example, if a question prompts users to enter their name, you can start the following questions with the answer.

- [Creating questions in bulk](#)

Create questions in bulk to speed up the development of your surveys.

- [Asking a question in a case](#)

You can use the Question shape to collect a single response from a user who is processing a case. By isolating individual questions, you can selectively capture user input without the formal structure of a survey.

- [Asking a group of related questions in a case](#)

You can use the Question page shape to collect information from a user who is processing a case. By providing a structured format for questions and answers, you can quickly incorporate user feedback into a case.

- [Running a survey in a case](#)

- [Best practices for creating a survey](#)

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 Survey**.
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 survey](#)

Adding a question to a question page

Add a question to a question page to prompt users for information in a format that you define.

1. In the header of Dev Studio, click **Configure Case Management Survey** .
2. On the **Surveys** tab, click the name of a survey.
3. Click a question page where you want to add the question, and then click **+Add question New** .
4. In the list that is displayed, select a type of question.
5. In the rich text editor, enter the text for your question.
6. Format your question, based on the type that you choose.

The following question types require formatting:

- [Slider](#)
- [Picklist](#)
- [Checklist](#)
- [Multi-text boxes](#)
- [Radio button matrix](#)

7. Click **Save survey**.

- [Formatting text boxes in a survey](#)

Format multiple text boxes in a survey to ask users to provide short text answers. For example, you can ask users to enter a first name or a phone number.

- [Formatting a picklist in a survey](#)

Format a picklist in a survey to define the choices that users can select to answer a question.

- [Formatting a checklist in a survey](#)

Format a checklist in a survey to define the choices that users can select to answer a question. For example, you can ask users to select the languages that they speak.

- [Formatting a slider in a survey](#)

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.

- [Formatting a radio button matrix in a survey](#)

Format a radio button matrix in a survey to ask a series of related questions that are answered by selecting one option from a repeating group of radio buttons. 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.

- [Creating questions in bulk](#)
- [Creating a stand-alone question](#)

Formatting text boxes in a survey

Format multiple text boxes in a survey to ask users to provide short text answers. For example, you can ask users to enter a first name or a phone number.

Before you begin: Add a question of type **Multi-text boxes** to your survey. For more information, see [Adding a question to a question page](#).

1. Find a multi-text boxes question to format, if you do not have one open already.
 - a. In the header of Dev Studio, click **Configure Case Management Survey** .
 - b. Click the name of a survey, and then click a question of type **Multi-text boxes**.
2. In the text field, enter the text of the question.
3. Label the text boxes.
 - a. In the **Textbox options** section, click **+Add value**.
 - b. In the **Display value** field, enter a label for the text box.
4. **Optional:** To control when the text box is displayed, define conditions.
 - a. Click **More Advanced view** .
 - b. From the **Visible when** drop-down list, select **When**, then click **Edit**.
 - c. From the **Field** drop-down list, select the field that controls whether the check box is displayed in the group.
 - d. From the **Comparator** drop-down list, select the comparator for the when condition.
 - e. Click **Select values**, and then select the values that determine whether the check box is displayed.
 - f. Click **Submit**.
 - g. Click **OK**.
5. Click **Save survey**.

- [Designing surveys](#)

Formatting a picklist in a survey

Format a picklist in a survey to define the choices that users can select to answer a question.

Before you begin: Add a question of type **Picklist** to your survey. For more information, see [Adding a question to a question page](#).

1. Find a picklist question to format, if you do not have one open already.
 - a. In the header of Dev Studio, click **Configure Case Management Survey** .
 - b. Click the name of a survey, and then click a question of type **Picklist**.
2. Define a value for each choice in the picklist.
 - a. In the **Picklist options** section, click **+ Add value**.
 - b. In the field that is displayed, enter a label for the choice.
3. **Optional:** Store the selected choice in a format that is different from the label.

For example, in a list of color schemes, you can use the hexadecimal color code to find similar colors from different manufacturers.

 - a. In the **Picklist options** section, click **More Advanced view** .
 - b. In the **Reporting value** field, replace the default value with your custom value.
4. **Optional:** To control which choices are available when users take the survey, define conditions.

For example, you can remove snow-related choices from a list of hobbies when a user lives in a tropical climate.

- a. In the **Picklist options** section, click **More Advanced view** .
- b. In the **Visible when** list, select **When**, and then click **Edit**.
- c. In the **Field** list, select a question in your survey.
- d. In the **Value** field, enter a value that is the same format as the expected answer to the question.
- e. In the **Comparator** list, select an operation that compares the answer to the question to the value that you provide.

- f. Click **OK**.
5. Click **Save survey**.

What to do next: Continue to manually manage your picklist, or create a data page to support dynamically sourced choices.

- [Dynamically sourcing the choices in a picklist](#)

Use a data page that determines the choices in a picklist to make your survey easier to maintain.

- [Designing surveys](#)
- [Dynamically sourcing the choices in a picklist](#)

Dynamically sourcing the choices in a picklist

Use a data page that determines the choices in a picklist to make your survey easier to maintain.

Before you begin: [Create a data page that defines the choices in a picklist](#).

1. Find and open your data page, by searching for it or by using the Records Explorer.
2. Click the **Definition** tab.
3. In the **Data sources** section, click the **Open** icon next to the source, such as a data transform, that defines the choices for your survey.
4. Review the configuration of the data source to identify which property stores the value for each choice.
5. Associate the data page with a picklist question in your survey.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. Click the name of your survey.
 - c. Click a **Picklist** type question so that it has focus.
 - d. In the **Picklist options** section, click **More Data page**.
 - e. In the **Data page** field, press the Down Arrow key, and then select your data page.

If the data page supports parameters, you can pass values for these parameters in the **Value** field.

- f. In the **Property for value** field, press the Down Arrow key, and then select the property that you identified in step 4.
 - g. **Optional:** To change the default labels for the choices, enter a property from the data page in the **Property for display text** field.
6. Click **Save survey**.

- [Formatting a picklist in a survey](#)
- [Finding rules by type](#)

Formatting a checklist in a survey

Format a checklist in a survey to define the choices that users can select to answer a question. For example, you can ask users to select the languages that they speak.

Before you begin: Add a question of type **Checklist** to your survey. For more information, see [Adding a question to a question page](#).

1. Find a checklist question to format, if you do not have one open already.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. Click the name of a survey, and then click a question of type **Checklist**.
2. Define the choices in the checklist.
 - a. In the **Checklist options** section, click **+ Add value**.
 - b. In the text field that is displayed, enter a label for the check box.
3. **Optional:** To control when a choice is included in the checklist, define conditions.
 - a. Click **More Advanced view**.
 - b. From the **Visible when** drop-down list, select **When**, then click **Edit**.
 - c. From the **Field** drop-down list, select the field that controls whether the check box is displayed in the group.
 - d. From the **Comparator** drop-down list, select the comparator for the when condition.
 - e. Click **Select values**, and then select the values that determine whether the check box is displayed.
 - f. Click **Submit**.
 - g. Click **OK**.
4. **Optional:** Store the selected choice in a format that is different from the label.
 - a. Click **More Advanced view**.
 - b. In the **Reporting value** field, replace the default value with your custom value.
For example, you in a list of languages that users speak, you can use an abbreviation for each language.
5. Click **Save survey**.

- [Designing surveys](#)

Formatting a slider in a survey

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.

Before you begin: Add a question of type **Slider** to your survey. For more information, see [Adding a question to a question page](#).

1. Find a slider question to format, if you do not have one open already.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. Click the name of a survey, and then click a question of type **Slider**.
2. In the text field, enter the text of the question.
3. In the **Question settings** section, format the slider.
 - a. In the **Starting range** field, enter the minimum numerical value for the scale.
 - b. In the **Ending range** field, enter the maximum numerical value for the scale.
 - c. In the **Interval** field, enter the numerical interval between the possible values on the scale.
 - d. **Optional:** To preset the answer to the question, enter a value in the **Default value** field.
4. Click **Save survey**.

- [Designing surveys](#)

Formatting a radio button matrix in a survey

Format a radio button matrix in a survey to ask a series of related questions that are answered by selecting one option from a repeating group of radio buttons. 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.

Before you begin: Add a question of type **Radio button matrix** to your survey. For more information, see [Adding a question to a question page](#).

1. Find a radio button matrix question to format, if you do not have one open already.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. Click the name of a survey, and then click a question of type **Radio button matrix**.
2. In the text field, enter the text of the question.
3. For each column in the radio button matrix, perform the following actions:
 - a. Click **+Add column**.
 - b. In the text field, enter the text that is displayed as a header of the column.
4. For each row in the radio button matrix, perform the following actions:
 - a. Click **+Add row**.
 - b. In the text field, enter the text that is displayed as a label of the row.

5. **Optional:** Set the scoring for each column by completing the **Question settings** section.
For example, you can set the higher score for the most severe level of pain.
6. **Optional:** Store the selected value in a column in a format that is different from the label.
 - a. Click **More Advanced view**.
 - b. In the **Reporting value** field, replace the default value with your custom value.
 For example, you can label the columns with adjectives describing different levels of pain, and then store the results as numbers.
7. **Optional:** To control when the row is displayed, define conditions.
 - a. Click **More Advanced view**.
 - b. From the **Visible when** drop-down list, select **When**, then click **Edit**.
 - c. From the **Field** drop-down list, select the field that controls whether the row is displayed in the matrix.
 - d. From the **Comparator** drop-down list, select the comparator for the when condition.
 - e. Click **Select values**, and then select the value that determines whether the row is displayed.
 - f. Click **Submit**.
 - g. Click **OK**.
8. Click **Save survey**.

- [Designing surveys](#)

Creating a stand-alone question

Create a stand-alone question, which you can reference outside the context of a survey, to support information gathering when a case is offline.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
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.
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. From the **Question layout** list, select an option that indicates how the text of the question is sourced in your application.
For example, you can define your own text or reuse the text from a paragraph rule.
7. Provide the source of the question text by entering a value in one of the available fields.
You can choose from:

Paragraph

A paragraph rule in your application.

Section reference

A section rule in your application.

Question

Plain text that you provide.

Question text

Rich text that you provide.

8. Format the fields that users interact with to answer the question.
 - For **Text area**, in the **Answer presentation settings** section, complete the following fields:

Default

Default text.

Rows

Height of the text area.

Columns

Width of the text area.

- For **Text box**, in the **Answer presentation settings** section, complete the following fields:

Size of text box

Maximum character count of the answer.

Default

Default date.

- For **Date or date/time**, in the **Answer presentation settings** section, complete the **Default** field by selecting the default date.
- For **Predefined list**, from the **Drop-down type**, select **Data page**. Complete the following fields:

Data page

Reference to the data page that returns a list of options.**Note:** You can use the **Value** field to pass parameters supported by the data page.

Property for value

Reference to the property on the data page that stores the options for the list.

Property for display text

Reference to the property from the data page that stores list item labels

- For **Predefined radio buttons**, from the **Drop-down type**, select **Data page**. Complete the following fields:

Data page

The data page that returns a list of options.**Note:** You can use the **Value** field to pass parameters supported by the data page.

Property for value

The property on the data page that stores the options for the list.

Property for display text

The property from the data page that changes the default list item labels.

- For **Custom list**, from the **Drop-down type**, select **Prompt list**. Complete the following actions:
 1. For each choice, click the **Add row** icon and complete the following fields:

When

A when condition that controls whether the choice is included in the list.

Display value

Choice label.

Value

The value copied to the Clipboard when the user selects this choice.

2. When you finish configuring all the choices, in the **Selected** column, click one option to set the default choice in the list.

- For **Custom radio buttons**, from the **Radio button source** list, select **Prompt list**. Perform the following actions:
 1. For each radio button, click the **Add row** icon and complete the following fields:

When

A when condition that controls whether the radio button is displayed.

Display value

Radio button label.

Value

The value that is copied to the Clipboard when the user selects this radio button.

2. When you finish configuring all the radio buttons, in the **Selected** column, click one option to set the default radio button.

- For a **Custom section**, perform the following actions:
 1. In the **Section reference** field, press the **Down Arrow** key and select the section that displays fields for the user to enter a response.
 2. In the **Param value** field, enter a corresponding value for each parameter that the section supports.
 3. In the **Map to property** field, press the **Down Arrow** key and select the name of a property that stores the answer that the user provides.**Note:** Some types of questions, such as custom sections, do not support property mapping.

4. Close the **Add question** dialog box by clicking **Save**.

What to do next:

Integrate your stand-alone question with a case by calling an API or adding a step to the case life cycle.

- [Asking a question in a case](#)
- [Requirements for surveys in offline mode](#)
- [Offline capability](#)

Configuring advanced options for a question page

Adjust question pages to meet your specific business needs by configuring advanced options. For example, categorize question pages to efficiently organize future reports, or to add an activity that performs an action before displaying the question page.

Before you begin: To categorize question pages, create field values of a *pyCategory* type. For more information, see [Creating a field value](#).

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click the **Question pages** tab, and then perform one of the following actions:
 - To create a question page, click **Create question page**, and then provide a name for your question page.
 - To edit a question page, select the question page.
3. In the **Edit question page** window, click **Advanced question page settings**.
4. In the **Category** field, press the Down arrow key, and then select a category for the question page.
5. In the **Validation** field, press the Down arrow key, and then select a validation rule for the question page.
6. In the **Pre-action** field, press the Down arrow key, and then select the activity that you want to invoke before the question page is displayed.
7. Click **Save**.

- [Adding a question to a question page](#)
- [Fine-tuning your survey](#)

Reusing answers in survey questions

To create more personalized and comprehensive surveys, reuse answers to questions in the subsequent questions. For example, if a question prompts users to enter their name, you can start the following questions with the answer.

Note: You can only reuse answers to questions that already exist in your survey or in a base class where you create your survey.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click the name of a survey.
3. Click **Add survey item New question**.
4. From the drop-down list, select the type of question that you want to create.
5. In the text box, enter a question, and then place the cursor where you want to reuse the answer.
6. Click the **Insert Property** icon.
7. Select a question whose answer you want to reuse.

The question that you select must appear in the survey before the question that you are creating.
8. **Optional:** To reuse a property from outside your current survey, select the **Relevant properties** check box.

Note: To use this option, you must enable a *pyEnableSurveyAdvancedSettings* when rule.

- [Adding a question to a question page](#)

Creating questions in bulk

Create questions in bulk to speed up the development of your surveys.

- [Populating a question template](#)

Populate a question template to define the single-answer questions that you want to create in bulk.

- [Importing a question template](#)

Import a question template to automate the creation of single-answer questions.

- [Troubleshooting question-template errors](#)

You can use a question template to create or update survey questions in bulk.

- [Adding a question to a question page](#)
- [Troubleshooting question-template errors](#)

Populating a question template

Populate a question template to define the single-answer questions that you want to create in bulk.

1. Download the question template.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. On the **Questions** tab, click **Import questions**.
 - c. Click **Download template** to save a Microsoft Excel file to your local machine.
 - d. Open the file.
2. For each question to create in bulk, populate a row in the file.
 - a. In the **ID** column, enter text that uniquely identifies the question.

If you enter the ID of an existing question, your application prompts you to update or skip the question when you import the question template.

- b. In the **Name** column, enter a label for the question.
3. Provide text for each question.
 - To use plain text:
 1. In the **Question Layout** column, select **plain**.
 2. In the **Question Plain Text** column, enter the question text.
 - To use styled text:
 1. In the **Question Layout** column, select **rte**.
 2. In the **Rich text question content** column, enter question text that includes HTML tags.

For example, you can use the `<em>` tag to italicize your question text.
 4. Configure the answer format for each question.
 - Attachment:
 - In the **Answer Mode** column, select **File**.
 - Text area, Text box, Date, or Date/time:

1. In the **Answer Mode** column, select **freeform**.
2. In the **Answer Freeform Format** column, select an option that corresponds to your answer format.
- Check box:
 - In the **Answer Mode** column, select **checkbox**.
- Radio buttons:
 1. In the **Answer Mode** column, select **radio**.
 2. In the **Answer option labels** column, enter a comma-separated list of the choices that users can select.
 3. In the **Answer option values** column, enter a comma-separated list of the values that map to the choices that users can select.
 4. In the **Radio button label orientation** column, choose whether the choices are stacked or side-by-side.
5. **Optional:**

To control how much screen space is used by horizontal radio buttons, enter an integer in the **Number of items before wrap** column.
- List:
 1. In the **Answer Mode** column, select **dropdown**.
 2. In the **Answer option labels** column, enter a comma-separated list of the choices that users can select.
 3. In the **Answer option values** column, enter a comma-separated list of the values that map to the choices that users can select.
- Custom section:
 1. In the **Answer Mode** column, select **section**.
 2. In the **Section Reference** column, enter the name of a section that is in the same class path as the question.
5. In the **Answer layout direction** column, choose whether the question and answer are stacked or side-by-side.
6. To change where your application stores the answer to a question on the clipboard, enter a property name in the **Answer Property** column that is in the same class path as the question.
7. To validate the answer that users provide at run time, enter the name of an edit validate rule in the **Validation** column that is in the same class path as the question.
8. Save and close the file but do not change the file format.

Result: Advanced functionality, such as conditional answers, is not supported by the question template. For more information about using advanced functionality after you import questions, see [Fine-tuning your survey](#).

- [Troubleshooting question-template errors](#)

Importing a question template

Import a question template to automate the creation of single-answer questions.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. On the **Questions** tab, click **Import questions**.
3. Click **Browse**.
4. Navigate to the location of your question template, and then click **Open** to upload the file.
5. **Optional:** To make the questions compatible with surveys that are outside your work pool, change the default class name in the **Apply to class** field.
6. Click **Next** to validate the entries in your question template.
7. Click **Import**.
8. Click **View details** to review the summary of questions that your application created, updated, or skipped.
9. Click **Done**.

What to do next: Add your questions to a survey or a question page in a survey.

- [Troubleshooting question-template errors](#)

Troubleshooting question-template errors

You can use a question template to create or update survey questions in bulk.

1. Perform one of the following actions:
 - If the template contains the IDs of questions that are in your application, go to step [2](#).
 - If the template has an invalid file structure or an invalid file type, go to step [3](#).
 - If questions in the template fail validation, go to step [4](#).
2. If the template contains the IDs of questions that are in your application, perform the following actions:
 - a. In the **Duplicate questions** section of the **Import survey questions** page, click **View details**.
 - b. Review the list of questions, to confirm that you intentionally added them to the template.
 - c. Decide how to process the duplicate questions.
 - To exclude them 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 the information from the first row that it finds only.
 - d. Click **Import**.
 - e. Click **Done**.
3. If the template has an invalid file structure or an invalid file type, perform the following actions:
 - a. Click **Download template** to save a Microsoft Excel file to your local machine.
 - b. Open the file.
 - c. Copy the information from your original template to the relevant columns in the file.
 - d. Save the file but do not change its format.
 - e. Click **Browse**.
 - f. Navigate to the location of your new question template, and then click **Open** to upload the file.
 - g. Click **Next**.
 - h. Click **Import**.
 - i. Click **Done**.
4. If questions in the template fail validation, perform the following actions:
 - a. In the **Failed validation** section of the **Import survey questions** page, click **View details**.
 - b. Review the error messages for each question to determine which columns have invalid data or reference an invalid rule. For example, you must provide the name of a section rule when the answer format for a question is set to **Section**.
 - c. Click **Back**.
 - d. Update your template to provide the required information.
 - e. Create the required rules in the same class path as the questions.
 - f. Click **Browse**.
 - g. Navigate to the location of your updated question template, and then click **Open** to upload the file.
 - h. Click **Next**.
 - i. Click **Import**.
 - j. Click **Done**.

- [Creating questions in bulk](#)

Asking a question in a case

You can use the Question shape to collect a single response from a user who is processing a case. By isolating individual questions, you can selectively capture user input without the formal structure of a survey.

1. Add the Question shape to the life cycle of your case.
 - a. On the **Workflow** tab, click **Life cycle**.
 - b. Hover over a process in a stage, and then click **+ Step More Utilities Question** .
 - c. Click **Select**.

Tip: For advanced processes, click **Configure process** to add the shape to the flow canvas.
2. Choose a question to ask the user.
 - To reuse an existing question:
 1. In the **Question type** list, select **Rule**.
 2. In the **Question rule** field, press the Down Arrow key and select the name of a question.

Complex questions are excluded from the list because they are supported in surveys only.
 3. Review the information in the **Question info** section to verify that the question text and answer format are relevant to the case.
 - To create a virtual question:

Use this approach when you are collecting a text response and the question is specific only to the process that contains the Question shape, because virtual questions are not reusable.

 1. In the **Question type** list, select **Text**.
 2. In the **Question text** field, enter the text of the question that is displayed at run time.
 3. In the **Map answer to property** field, press the Down Arrow key and select a property to store the value of the answer that the user provides.
 - To create a reusable question:

Use this approach when you are collecting a response in a different format, such as a check box, or plan to reuse the question in another process or survey.

 1. Create a stand-alone question.
 2. In the **Question type** list, select **Rule**.
 3. In the **Question rule** field, press the Down Arrow key and select the name of your question.
3. **Optional:** To ensure that the user provides an answer to the question, select the **Required** check box.

Some formats, such as read-only text, do not support this option.
4. **Optional:** To validate the answer that the user provides, define pre and post processing.
 - a. In the **Pre-question processing activity** and **Post-question processing activity** fields, press the Down Arrow key and select the name of an activity.

These activities are called when the user is online.
 - b. In the **Pre-question javascript function** and **Post-question javascript function** fields, enter the names of functions that are bundled and attached to the harness of the case.

These functions are called when the user is offline.
5. Click **Save**.

Asking a group of related questions in a case

You can use the Question page shape to collect information from a user who is processing a case. By providing a structured format for questions and answers, you can quickly incorporate user feedback into 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. On the **Workflow** tab, click **Life cycle**.
3. Hover over a process in a stage, and then click **+ Step More Utilities Question page** .
4. Click **Select**.
5. In the **Select question page** field, press the Down Arrow key, and then select the name of your question page.
6. **Optional:** Ask a specific user, instead of the current case worker, to answer the questions on the page.
 - a. In the **Route to** list, select **Specific user**.
 - b. In the **Operator** field, press the Down Arrow key and then select the name of a user in your application.
7. To control whether the question page is displayed in the case at run time, press the Down Arrow key in the **Display when** field, and then select a when condition.
8. Click **Save**.

Fine-tuning your survey

Fine-tune the way that your survey displays questions and evaluates answers to ensure that you collect valid information in a case.

Use the following techniques to fine-tune your survey:

- [Making a question required in a survey](#)

Make a question required to ensure that your survey collects complete information.
- [Conditionally displaying a question in a survey](#)

Define conditions to ensure that your survey displays relevant questions, based on the context of the survey participant.
- [Conditionally displaying a question page in a survey](#)

Define conditions to ensure that your survey displays related questions that are relevant, based on the context of the survey participant.
- [Validating answers in a survey](#)

Display an error message in your survey when an answer is invalid, to ensure that the information that users provide is accurate and in an expected format.
- [Displaying a warning in a survey](#)

Use a warning message to encourage survey participants to provide complete answers.
- [Changing the run-time order of questions in a survey](#)

Add a branch to a survey to define the conditions that skip or revisit questions at run time.
- [Maximizing screen space in a survey](#)

Change the way that your survey displays questions to maximize screen space and improve usability.

- [Adding supporting information to a survey](#)

Add supporting information to a survey to enhance the text of a question.

- [Adding supporting information to a survey](#)
- [Maximizing screen space in a survey](#)

Making a question required in a survey

Make a question required to ensure that your survey collects complete information.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click the name of a survey, and then click the question that requires an answer.
3. In the **Question settings** section, select the **Required** check box.

This option is not available for multi-answer questions, however, you can make a multi-answer question required by performing validation on the answers that the user provides at run time.

4. Click **Save survey**.

- [Validating answers in a survey](#)

Conditionally displaying a question in a survey

Define conditions to ensure that your survey displays relevant questions, based on the context of the survey participant.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click the name of a survey.
3. Click a question in the survey so that it has focus.
4. In the **Visibility conditions** section, click the **Gear** icon.
5. Define the criteria for displaying the question at run time.

Tip: To evaluate multiple conditions, click the **Add a row** icon.

- a. In the **Field** list, select a question in your survey.
- b. In the **Value** field, enter a value that is the same format as the expected answer to the question.

For example, you can enter a positive integer in this field when the answer represents years of experience.

- c. In the **Comparator** list, select an operation that compares the answer to the value that you provide.

- d. Click **OK**.

6. Click **Save survey**.

- [Conditionally displaying a question page in a survey](#)

Conditionally displaying a question page in a survey

Define conditions to ensure that your survey displays related questions that are relevant, based on the context of the survey participant.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click the name of a survey.
3. Click a question page in the survey so that it has focus.
4. In the **Visibility conditions** section, click the **Gear** icon.
5. Define the criteria for displaying the question page at run time.

Tip: To evaluate multiple conditions, click the **Add a row** icon.

- a. In the **Field** list, select a question in your survey.
- b. In the **Value** field, enter a value that is the same format as the expected answer to the question.

For example, you can enter a positive integer in this field when the answer represents years of experience.

- c. In the **Comparator** list, select an operation that compares the answer to the value that you provide.

- d. Click **OK**.

6. Click **Save survey**.

- [Conditionally displaying a question in a survey](#)

Validating answers in a survey

Display an error message in your survey when an answer is invalid, to ensure that the information that users provide is accurate and in an expected format.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click the name of a survey.
3. Click a question so that it has focus.
4. In the **Validations** section, click the **Gear** icon.
5. Define the validation conditions.

- a. In the **Message** field, enter text for an error message.
- b. In the **Field** list, select a question in your survey.
- c. In the **Value** field, enter a value that is the same format as the expected answer to the question.

For example, you can enter a positive integer in this field when the answer represents years of experience.

Tip: For more guidance which values you can evaluate in the condition, click **Select values**.

- d. In the **Comparator** list, select an operation that compares the answer to the value that you provide.

For example, you can verify that the user provides a phone number for each name on the emergency contact question page.

- e. Click **OK**.

6. Click **Save survey**.

- [Making a question required in a survey](#)
- [Displaying a warning in a survey](#)
- [Populating a question template](#)

Displaying a warning in a survey

Use a warning message to encourage survey participants to provide complete answers.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. On the **Questions** tab, click a question.
3. Expand the **Warnings** section.
4. Define the conditions that control whether the warning is displayed.

The types of conditions that you can use are based on the type of question.

- To use a when condition:

Use this approach when users complete your survey online.

1. In the **Visibility** list, select **Condition (when rule)**.
2. In the **Warning when** field, press the Down Arrow key and select the name of a when condition that evaluates property values in the survey.

- To use a Boolean expression:

Use this approach users complete your survey offline.

1. In the **Visibility** list, select **Condition (expression)**.
2. In the **Warning expression** field, enter a string expression that evaluates property values in the survey.

- To use a combination of when conditions and Boolean expressions:

1. In the **Visibility** list, select **Condition (expression)**.
2. Click the **Condition builder** icon to open the **Visible when** dialog box.
3. Enter values in the fields based on the type of conditions and property values to evaluate at run time.
4. Click **Save changes** to close the **Visible when** dialog box.

5. Click **Save**.

- [Validating answers in a survey](#)

Changing the run-time order of questions in a survey

Add a branch to a survey to define the conditions that skip or revisit questions at run time.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Click the name of a survey.
3. Click **+ Add survey item Logic**.
4. In the **Branch** field, click the **Edit** icon, and then enter a label for the branch.
5. In the **Branching conditions** section, define the conditions that determine the next item in the survey.
 - a. Click **Click to add condition**.
 - b. In the **Field** list, select a question in your survey.
 - c. In the **Value** field, enter a value that is the same format as the expected answer to the question.

For example, you can enter a positive integer in this field when the answer represents years of experience.

 - d. In the **Comparator** list, select an operation that compares the answer to the value that you provide.
 - e. **Optional:** To define more conditions, click **Add path**.
 - f. Click **OK**.
 - g. In the **Go to** list, select a survey item to display to the user when the condition returns a true value.
 - h. In the **Otherwise go to** list, select a survey item to display to the user when the condition returns a false value.

For example, you can skip to the next question page in a personal health survey, when users indicate that they do not smoke.
6. Click **Save survey**.

- [Designing surveys](#)

Maximizing screen space in a survey

Change the way that your survey displays questions to maximize screen space and improve usability.

Use any of the following techniques to maximize screen space:

- Use one screen for each part of the survey.
 - 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, click **Life cycle**.
 - c. Click a Questionnaire step.
 - d. In the **Navigation type** list, select **Plain**.
 - e. Select the **Run as screenflow** check box.
 - f. In the **Screenflow type** list, select an option to indicate how the questions are distributed across multiple screens.
 - g. To add navigation links to the screen, select the **Show bread crumb** check box.
- Hide supporting information that is unnecessary.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. On the **Question pages** tab, click the name of a question page.
 - c. Expand the **Advanced question page settings**.
 - d. Select the **Hide label** check box.
 - e. Select the **Hide instructions** check box.
 - f. Click **Save**.
- Display radio buttons in rows instead of columns.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. On the **Questions** tab, click the name of a question with **Question type** set to **Radio buttons**.
 - c. In the **Edit question** dialog box, expand the **Answer presentation settings** section.
 - d. In the **Display of radio buttons** list, select **Horizontal**.
 - e. **Optional:** To control how many radio buttons are displayed in each row, enter an integer in the **Number of items before wrap** field.
 - f. Click **Save**.
- Change the answer layout for each question to display the answer fields next to the question text instead of below the question text.
 - a. In the header of Dev Studio, click **Configure Case Management Survey**.
 - b. On the **Questions** tab, click the name of a question.
 - c. In the **Edit question** dialog box, expand the **Answer presentation settings** section.
 - d. In the **Answer layout** list, select **Question and Answer adjacent to each**.
 - e. Click **Save**.
- [Conditionally display questions](#).
- [Fine-tuning your survey](#)
- [Best practices for creating a survey](#)

Adding supporting information to a survey

Add supporting information to a survey to enhance the text of a question.

Use any of the following techniques to add supporting information to a survey:

- [Adding hover text to a question](#)

Add hover text to a question in a survey to provide assistance to users in a way that maximizes screen space.

- [Attaching a file or URL link to a question](#)

Attach a file or URL link to a question to provide supporting information that helps users provide answers more quickly.

- [Displaying a block of text in a question page](#)

Display a block of read-only text on a question page in a survey to provide supporting information that helps users provide answers more quickly.

- [Designing surveys](#)
- [Fine-tuning your survey](#)

Adding hover text to a question

Add hover text to a question in a survey to provide assistance to users in a way that maximizes screen space.

1. Create a section in a class and ruleset that is compatible with your survey, and then configure the section to display text.

For example, you can add a paragraph to a section that defines the different types of medical plans.

For more information about creating sections, see [Presenting content in sections](#).

2. In the header of Dev Studio, click **Configure Case Management Survey**.
3. On the **Questions** tab, click a question.
4. Select the **Include smart info** check box.
5. In the **Smartinfo section** field, press the Down Arrow key and then select the name of your section.
6. Click **Save**.

- [Fine-tuning your survey](#)
- [Displaying a block of text in a question page](#)
- [Attaching a file or URL link to a question](#)

Attaching a file or URL link to a question

Attach a file or URL link to a question to provide supporting information that helps users provide answers more quickly.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. On the **Questions** tab, click a question.
3. Select the **Include supporting document** check box.
4. In the **Supporting label** field, enter text to display in the tool tip for the file or URL.
5. Attach the supporting information.
 - To attach a file:
 1. In the **Supporting type** list, select **File**.
 2. Click **Choose File** and navigate to a file on your local machine.
 3. Click **Upload File**.
 - To attach a URL link:
 1. In the **Supporting type** list, select **URL**.
 2. In the **Supporting URL** field, enter the address of a website or supporting document.

6. Click **Save**.

Result:

At run time, an icon is displayed next to the question. Users can click the icon to open the supporting website or document.

- [Fine-tuning your survey](#)
- [Displaying a block of text in a question page](#)
- [Adding hover text to a question](#)

Displaying a block of text in a question page

Display a block of read-only text on a question page in a survey to provide supporting information that helps users provide answers more quickly.

1. In the header of Dev Studio, click **Configure Case Management Survey**.
2. Create the block of text.
 - a. On the **Questions** tab, click **Create question Browse all types Single answer Info text**.

Although users cannot respond to a block of read-only text in a survey, you must create a question to store the text.

- b. In the **Name** field, enter a unique label for the question.
 - c. In the **Question text** field, enter the content for the block of text.
 - d. Click **Save**.
3. Add the question to a question page in your survey.
 - a. On the **Surveys** tab of the Survey Management landing page, click the name of a survey.
 - b. Click a question page in the survey.
 - c. Click + **Add question From library**.
 - d. Click + **Add** next to the question that defines the block of text, and then click **Add**.
 4. Click **Save survey**.

What to do next:

You can change the position of the block of text by reordering the questions on the question page.

- [Fine-tuning your survey](#)
- [Populating a question template](#)
- [Adding hover text to a question](#)
- [Attaching a file or URL link to a question](#)

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.

- [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.

- [Controlling access to case attachments](#)

Control access to the attachments in a case to enforce security requirements and prevent case workers from inadvertently modifying case data.

- [Managing work across your team](#)

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.

- [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.

- [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.

- [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.

- [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.

- [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.

- [Access Manager](#)

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.

Before you begin: To manage case types on the Work & Process tab of Access Manager, you must be logged in as an operator associated with an access group that contains the *PegaRULES:SecurityAdministrator* or *PegaRULES:BasicSecurityAdministrator* role.

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](#)
- [Configuring case type access](#)

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.

Before you begin: To manage case types on the Work & Process tab of Access Manager, you must be logged in as an operator associated with an access group that contains the *PegaRULES:SecurityAdministrator* or *PegaRULES:BasicSecurityAdministrator* role. **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. **Result:** 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](#)
- [Configuring case type access](#)

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.

Before you begin: To manage case types on the Work & Process tab of Access Manager, you must be logged in as an operator associated with an access group that contains the *PegaRULES:SecurityAdministrator* or *PegaRULES:BasicSecurityAdministrator* role.

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](#)
- [Configuring case type access](#)

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.

Before you begin: To manage case types on the Work & Process tab of Access Manager, you must be logged in as an operator associated with an access group that contains the *PegaRULES:SecurityAdministrator* or *PegaRULES:BasicSecurityAdministrator* role. 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](#)
- [Configuring case type access](#)

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.

Before you begin: To manage case types on the Work & Process tab of Access Manager, you must be logged in as an operator associated with an access group that contains the *PegaRULES:SecurityAdministrator* or *PegaRULES:BasicSecurityAdministrator* role. **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](#)
- [Configuring case type access](#)

Controlling access to case attachments

Control access to the attachments in a case to enforce security requirements and prevent case workers from inadvertently modifying case data.

- [Categorizing case attachments](#)

Define business classifications for case attachments to provide access control policies that are specific to each attachment category.

- [Restricting user actions for case attachments](#)

Define conditions or privileges in an attachment category to control which actions a user can take on a case attachment.

Categorizing case attachments

Define business classifications for case attachments to provide access control policies that are specific to each attachment category.

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.
3. On the **Settings** tab, click **Attachment categories**.
4. Click **+ Add attachment category**.
5. 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.
6. **Optional:** To ensure that there is at least one attachment in this category when a user submits a case, select the **Required for resolution** check box.
7. Click **OK**.
8. Click **Save**.

Result: Your changes are applied to all unresolved cases. Users can select the attachment category when they add an attachment to a case.

- [Configuring and working with optional actions in case types](#)

Restricting user actions for case attachments

Define conditions or privileges in an attachment category to control which actions a user can take on a case attachment.

Before you begin: You must create at least one attachment category for a case type or a specialize a standard attachment category from the *Work-* class.

1. In the navigation panel, 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 types of case attachments 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 by using any combination of the following techniques.
 - To restrict user actions by privilege:
 1. Click **+ Add privilege**.
 2. In the **Privilege** column, press the Down Arrow key, and then select the name of a privilege.
 3. Select a check box in one or more columns, based on the user operations that this privilege grants.
For example, select the **Delete own** check box to allow users to delete the attachments that they create.
 - To restrict user actions by when conditions:
 1. Click **+ Add when**.
 2. In the **When** column, press the Down Arrow key, and then select the name of a when condition.
 3. Select a check box in one or more columns, based on the user operations that this when condition grants.
For example, select the **Create** check box to allow users to add attachments to a case.
 4. Repeat steps a through c by selecting the *Never* when condition for user operations that are denied.
For example, if you define a condition that allows users to create attachments, you must define another condition that restricts editing, viewing, and deleting attachments. The absence of a privilege or when condition does not automatically restrict a user operation.
6. **Optional:** To allow users to choose which teams can access the attachments that they provide, select the **Enable attachment-level security** check box.
7. Click **Save**.

Result:

A user can perform an action only when all when conditions return a true value, the user belongs to the required team, and the user holds at least one of the required privileges.

What to do next: Grant a privilege to case workers who resolve cases that includes create permissions for this attachment category.

- [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.

- [Attachment types](#)
- [Requiring attachments for stage entry](#)
- [Configuring and working with optional actions in case types](#)
- [Creating a specialized or circumstance rule](#)

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 `proconfig.xml` file to change the default size limit of 1 GB for attachments.

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 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](#)