Pega Platform '24.2

3 September 2025

©2025 Pegasystems Inc

CONTENTS

Case Management 13
Introduction to Case Management 16
Identifying Case Type elements 19
Understanding Case hierarchy 25
Identifying people involved in Case Types 29
Building Case Types 33
Providing data for Cases 38
Understanding and complying with Case Management best practices 41
Automating work by creating Case Types 44
Case Types 46
Creating a Case Type with Autopilot 49
Creating a Case Type manually 54
Creating a Constellation Case Type in a traditional UI application 57
Reusing a Constellation Case Type from an existing application 59
Limitations when using Constellation with traditional UI 61
Creating a child Case Type in Case Designer 63
Creating child Case Types in Dev Studio 65
Creating a Case Type Rule form in Dev Studio 67
Remote Case Types 83
Configuring a consumer system for Remote Case Types 86
Configuring a producer system for Remote Case Types 90
Creating Remote Case Types 97
Updating connection details for Remote Case Types 103
Fetching a list of selected Remote Cases with a queryable Data Page 104
Error messages for Remote Case Types 108
Creating Cases with a web embed 114
Case Lifecycle elements 115
Stages in a Case Lifecycle 119
The Create Stage 123

©2025 Pegasystems Inc

The Create Stage FAQ. 125
Adding the Create Stage to existing Case Types. 129
Technical considerations for the Create Stage in existing Case Types 131
Creating a Primary Stage. 133
Creating an Alternate Stage. 136
Defining the entrance criteria for a Stage. 138
Configuring a Case resolution. 145
Defining Conditions for skipping a Stage. 148
Removing a Stage. 151
Best practices for Stages. 153
Processes in a Case Lifecycle. 156
Adding a sequential Process to a Stage. 159
Adding Steps to Processes. 161
Adding a parallel Process to a Stage. 164
Defining Conditions for starting a Process. 167
Conditionally starting a Process. 170
Configuring navigation to a previous Step. 174
Removing a Process from a Stage. 176
Draft mode of Case Processes. 177
Steps in a Case Lifecycle. 178
Step types. 181
Collecting information from a user. 185
Assigning tasks to users. 189
Assigning users automatically at run time with business logic. 191
Configuring custom business logic-based routing. 195
Configuring custom routing logic for Assignments. 198
Adding instructions to Assignments. 199
Adding a Multi-step Form to a Stage. 202
Requesting approval from users. 206
Requesting approval by email and push notifications. 211
Configuring cascading approvals. 216
Configuring approvals with an authority matrix. 220

PEGA ©2025 Pegasystems Inc

Adding an automated Step to a Process. 224
Attaching content to a Case. 226
Creating a PDF from Case data. 231
Creating child and Remote Cases. 235
Creating contextual Cases in traditional UI applications. 243
Collecting digital signatures in a Case. 246
Finding duplicate Cases. 248
Generating Case documents. 259
Case documents tags. 264
Case documents specifications. 271
Case documents error messages. 273
Activities for integrating Cases with Excel documents. 277
Tags for generating .docx documents through APIs. 285
Automatically moving a Case to a different Stage. 289
Pausing and resuming Processes in Cases. 291
Selective dependency check in the Wait shape configuration: use case. 303
Populating Case data with a Data Transform. 305
Populating Case data with personalized AI-generated information. . . 307
Posting messages to a Case. 309
Preloading a Data Page. 312
Propagating data between parent and child Cases. 314
Running a queued background process in a Case. 320
Running a questionnaire in a Case. 322
Sending event notifications from Cases. 327
Sending automatic emails from Cases. 330
Sending push notifications from Cases. 337
Saving data in a Data Page as part of a Case Lifecycle. 340
Spinning-off independent Cases from the current Case. 342
Extending a Data Model for a Case. 352
Creating fields for capturing data. 359
Primary Fields. 362

©2025 Pegasystems Inc

Configuring Primary Fields. 363
Embedding data in a Case. 366
Referencing a Case Type. 368
Referencing a data object. 375
Querying a Data Page. 377
Classifying Cases by predicting field values. 381
Designing the UI in Case Types. 388
Restricting user input in a field. 389
Validating input values on a form. 390
Validating field input in complex scenarios. 396
Adding a Validation Rule to a Flow Action. 399
Validating character patterns. 400
Configuring and tailoring Case Types. 401
Configuring Case Type settings. 402
Configuring general Case Type settings. 406
Selecting a license category for a Case Type. 406
Changing the Case icon. 409
Enabling autosave for forms. 412
Changing the Case ID format. 414
Tracking Case workers based on geolocation data. 416
Providing the Get Next Work link in a Case. 419
Configuring collaboration settings. 421
Enabling Case tags. 421
Enabling Case following. 422
Configuring Pulse for Case Types. 423
Enabling field-level auditing. 425
Packaging Rules and data that contain field-level auditing information. 428
The extension point for field-level auditing. 429
Displaying the index keys of audited properties in the field-level auditing layout. 429
Sharing data between parent and child Cases. 438
Initializing data at Case creation. 443

©2025 Pegasystems Inc

Initializing data at Case creation in Dev Studio. 446
Configuring Case creation. 447
Enabling automatic creation of child Cases. 447
Enabling automatic conditional creation of child Cases. 449
Enabling creation of Cases by email. 451
Email account limitations. 454
Creating a Case by using an API. 455
Defining default data sources. 457
Detecting bias in Case Decisions. 460
Managing concurrent access to a Case. 469
Overriding the locking strategy of a child Case. 473
Locking stand-alone Cases. 474
Notifying Case participants about Case events. 477
Sending email notifications about Assignments routed to a user's Worklist. 478
Sending push notifications about Assignments routed to a user's Worklist and missed SLAs. 482
Sending confirmations of completed Assignments. 483
Sending custom notifications about new Assignments. 484
Disabling Assignment notifications. 485
Predicting Case Outcomes. 486
Defining Case participants. 489
Adding Case participants to a Case Type. 494
Using Data Transforms for automatic creation and prepopulation of Case participants. 498
Case participants configuration in the Work Parties Rule form. 504
Managing related Case Types. 509
Configuring Case-wide validation for correct data entry in fields. 510
Including indexed data during Case search. 515
Case archival settings. 515
Configuring complex Processes. 516
Creating a stand-alone Process. 519

©2025 Pegasystems Inc

Process shapes. 522
Connecting shapes in a Process. 540
Shape icons in a Process. 546
Types of Processes. 551
Assignment shapes in Processes. 557
Annotating a Process. 562
Keyboard shortcuts for Processes. 563
Creating a Screen Flow in Dev Studio. 565
Changing the path of a Process. 571
Adding Decisions to Processes. 571
Types of decision logic. 574
Changing the decision logic in a Process. 575
Adding a conditional path to a Process. 577
Adding a conditional path to an Assignment. 578
Adding a results-based path to a Process. 580
Subprocesses. 581
Calling one Process from another Process. 588
Adding additional configuration to a Subprocess. 589
Running multiple instances of different Subprocesses with a Split Join shape. 593
Running multiple instances of the same Subprocess with a Split for Each shape. 598
Flow Actions. 603
Creating Flow Actions. 607
Defining the run-time presentation of Flow Actions. 609
Configuring pre-processing for a Flow Action. 615
Configuring post-processing for a Flow Action. 618
Configuring back-to-back processing for a Flow Action. 622
Configuring Action buttons for a Flow Action. 625
Configuring indicator for a Flow Action. 630
Adding optional Actions to paused Processes. 632
Managing Case Types. 634

©2025 Pegasystems Inc

Importing a Case Type. 634
Renaming Case Types. 636
Deleting Case Types and Cases. 638
Automated and intelligent work assignment with Get Next Work. 639
Enabling Get Next Work button on a landing page. 646
Get Next Work customization. 648
Enabling bulk processing of Cases in Constellation. 657
Changing Case Statuses. 664
Case Status values. 666
Defining a Case Status. 668
Standard Rules and extension points for Case Status. 673
Changing a Stage with open Assignments. 674
Completing work on time. 677
Urgency. 682
Setting SLAs for Cases. 686
Setting SLAs for Stages, Processes, and Steps. 689
Setting SLAs for ad hoc Assignments. 692
Adjusting initial Urgency for an Assignment. 695
Creating an SLA Rule. 697
Configuring delayed service-level processing. 700
Defining an escalation Action for a Case. 702
Defining an escalation Action for a Stage and Process. 704
Defining an escalation Action for an incomplete Assignment. 705
Engaging and notifying stakeholders. 707
Creating a Notification Rule. 708
Notification delivery Channels. 711
Configuring a Web gadget notification delivery Channel for Portal notifications. 713
Configuring an email notification delivery Channel. 715
Configuring a mobile notification delivery Channel. 717
Configuring additional parameters for notifications delivery Channels. . . 718
Displaying application-specific correspondence. 719

PEGA ©2025 Pegasystems Inc

Creating a notification Channel. 722
Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools. 733
Pega GenAI Coach in Case Management. 733
Enabling Pega GenAI Coach in a Case Type or Data Object. 737
Creating a new Pega GenAI Coach. 740
Setting definitions for Pega GenAI Coach. 742
Adding a data source to a Pega GenAI Coach. 746
Selecting a generative AI model for a Coach. 750
Adding a Pega GenAI Coach as a widget. 751
Responding to business exceptions in a workflow. 753
Creating a ticket. 755
Policy overrides and suspended work items. 756
Configuring and working with optional Actions in Cases. 759
Adding optional Actions to Cases. 760
Adding optional Actions to Stages. 762
Displaying optional Actions conditionally. 763
Adding supporting optional Processes to Cases. 766
Adding supporting optional Processes to Stages. 768
Displaying supporting Processes Conditionally. 770
Preparing for collaboration with users by using Pulse. 773
Adding the Pulse gadget to your application. 775
Enabling users to post messages in the activity feed. 776
Creating feed sources for activity feeds. 778
Adding feed sources to activity feeds. 779
Determining default feed sources. 781
Configuring the display of feed sources. 782
Configuring Pulse email notifications. 785
Configuring Pulse feed sources for Constellation. 793
Markdown in Pulse. 794
Integrating Cases with external resources. 796
Integrating a Case with an external resource. 797

PEGA ©2025 Pegasystems Inc

Assigning a task to a robotic Work Queue. 797
Assigning a task to an external application. 803
Connecting a Process to an external application. 807
Running a Decision strategy from a Process. 810
Calling an activity or Automation from a Process. 811
Enabling Cases for creation through web and chat Channels. 815
Collecting information from an external user. 816
Configuring Directed Web Access. 818
Sourcing attachments from external storage. 825
Storing attachments in an external storage. 838
Designing questionnaires. 846
Creating a questionnaire. 848
Adding a question page to a questionnaire. 849
Adding a question to a question page. 852
Formatting text boxes in a questionnaire. 857
Formatting a picklist in a questionnaire. 860
Formatting a checklist in a questionnaire. 863
Formatting a slider in a questionnaire. 866
Formatting a radio button matrix in a questionnaire. 869
Formatting a currency question in a questionnaire. 872
Formatting a decimal question in a questionnaire. 875
Scoring answers in a questionnaire. 877
Best practices for creating questionnaires. 880
Running a questionnaire in a Case. 881
Fine-tuning your questionnaire. 886
Making a question required in a questionnaire. 887
Conditionally displaying a question in a questionnaire. 889
Conditionally displaying a question page in a questionnaire. 890
Displaying questions in a read-only mode. 892
Validating answers in a questionnaire. 894
Displaying a warning in a questionnaire. 896
Changing the run-time order of questions in a questionnaire. 898

PEGA ©2025 Pegasystems Inc

Displaying a block of text in a question page. 901
Visualizing questionnaire data by creating Insights in Constellation. 904
Using advanced questionnaire features in Dev Studio. 905
Creating a stand-alone question with a single answer. 906
Creating a stand-alone question with multiple answers. 911
Reusing answers in questions. 915
Creating questions in bulk. 917
Populating a question template. 918
Importing a question template. 922
Troubleshooting question template errors. 923
Asking a question in a Case. 925
Asking a group of related questions in a Case. 929
Creating reports based on questionnaires in Dev Studio. 931
Securing Case access. 932
Configuring Case Type access. 933
Viewing authorizations for Case Type items in a single Access Group. . . . 934
Editing authorizations for Case Type items in a single Access Group. 935
Viewing authorizations for Case Type items in all Access Groups. 936
Viewing authorizations for Processes and Flow Actions in a single Access Group. 938
Editing authorizations for Processes and Flow Actions in a single Access Group. 939
Controlling access to Case attachments. 940
Categorizing Case attachments. 941
Attachment types. 942
Restricting user actions for Case attachments. 944
Extension points and supporting Rules for attachments. 947
Best practices for compliance and updatability. 951
Fixing Cases in flight with Problem Flows. 956
Troubleshooting shape-related configuration issues for Cases in flight. . . . 962
Interacting with Cases at run time. 966
Creating and managing Cases. 966

PEGA ©2025 Pegasystems Inc

Bulk processing Cases. 969
Creating ad hoc Assignments. 983
Creating ad hoc Cases. 986
Creating Cases from tables. 988
Creating Cases from the navigation pane of a Portal. 993
Creating Cases from the Actions menu of another Case. 995
Following Cases. 997
Managing Case followers. 999
Interacting with parent and child Cases on multiple tabs. 1000
Out-of-the-box Portal features for creating and managing Cases. 1002
Reopening resolved Cases. 1008
Setting notification preferences. 1011
Tagging in Cases. 1015
Transferring Assignments. 1024
Using Insights in Case Management. 1025
Collaborating on Cases. 1029
Collaborating with users by using Pulse. 1029
Posting a message in Pulse. 1030
Replying to a message in Pulse. 1037
Restricting Case references in Pulse. 1044
Tagging a message in Pulse. 1045
Managing Pulse messages. 1046
Creating an Assignment in Pulse. 1049
Collaborating with users by using Spaces. 1051
Collaborating on shared content by using documents. 1082
Creating a document in an application. 1085
Discussing and managing a document. 1089

PEGA ©2025 Pegasystems Inc

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 Assignments into Stages, you can easily monitor how and when the Assignments are completed.

The following video provides the overview of Case Management in Pega Platform:

Case Management core elements

A Case, which represents a business process, is made up of many Stages, Processes, Assignments, policies, and supporting content. The Case as a whole continues to change throughout its Lifecycle due to internal and external events. Depending on the context of the Case, individual Assignments, Processes, or Stages can be interacted with by various stakeholders. 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 stakeholders work with Case timelines and relevant context, such as documents and attachments. Case Management facilitates resolving business processes, and guiding stakeholders to the information and tasks most relevant to their roles in the Case. With personalized event notifications, spaces, and Pulse, stakeholders can collaborate, ensuring that every Case worker is well-informed about a progress in a Case.

The following figure shows a visual representation of a business process with Stages, Processes, and Steps that can result in approving or rejecting a job candidate:

©2025 Pegasystems Inc

A business process of reviewing job candidates presented in Case Designer

Blueprint for optimizing Case Type development process

As a best practice, use Pega GenAI Blueprint™ to develop the foundations of a new application. Describe your goals, and Pega GenAI Blueprint will instantly suggest optimal Case Type designs, based on GenAI and a vast repository of workflow best practices. By creating a Blueprint on https://pega.com/blueprint, and then importing the Blueprint to Pega Platform, you significantly accelerate the Case Type development process. The following figures show a sample use of Pega GenAI Blueprint to generate a Case Type in a car insurance application:

©2025 Pegasystems Inc

Generating a Case Lifecycle using Pega GenAI Blueprint

For more information, see Creating a new application from a Blueprint.

You can also create a Case Type manually or by using Pega GenAI Autopilot™. For more information, see Creating a Case Type manually and Creating a Case Type with Autopilot.

- **Introduction to Case Management**
- **Automating work by creating Case Types**
- **Configuring and tailoring Case Types**
- **Completing work on time**
- **Engaging and notifying stakeholders**
- **Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools**

PEGA ©2025 Pegasystems Inc

- **Integrating Cases with external resources**
- **Designing questionnaires**
- **Securing Case access**
- **Best practices for compliance and updatability**
- **Fixing Cases in flight with Problem Flows**
- **Interacting with Cases at run time**

Related information

- Case Management

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.

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

PEGA ©2025 Pegasystems Inc

Visualization of a process of approving a job candidate

As your business circumstances change, you can add and remove Steps in the Process, add optional and supporting Actions, and, finally, manage data and Personas in the Process.

Pega Platform 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:

©2025 Pegasystems Inc

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 trial 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.

PEGA ©2025 Pegasystems Inc

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

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

- **Identifying Case Type elements**
- **Understanding Case hierarchy**
- **Identifying people involved in Case Types**
- **Building Case Types**
- **Providing data for Cases**
- **Understanding and complying with Case Management best practices**

Identifying Case Type 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

PEGA ©2025 Pegasystems Inc

use tools that help you achieve your business objectives in a flexible way, so that you can respond to dynamic and unique circumstances.

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

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

The outcome that you want to achieve

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

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

©2025 Pegasystems Inc

Outcomes of a Case Type

The paths that lead to the outcome

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

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

©2025 Pegasystems Inc

Paths towards resolution

The elements of your business process

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

©2025 Pegasystems Inc

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

Transition of work

The people involved in your business process

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

©2025 Pegasystems Inc

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

Different types of users and devices

The data that your business process requires

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

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

©2025 Pegasystems Inc

Data for a business process

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.

Related information

- App Studio overview

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

©2025 Pegasystems Inc

your business needs helps you make the most of the tools that Pega Platform™ provides and create the most efficient solutions.

For example, if your business objective is to review insurance claims, you might need to consider different scenarios while processing a claim. In a car accident insurance claim, a customer might file both body injury and vehicle damage claims. To process the Case, and to estimate the total amount of money that the customer should receive in compensation, you need to process the injury and damage Cases first. Using the results of these two Cases, you can calculate the final insurance payout. To achieve this goal, you can create an **Accident Claim** parent Case Type, and then **Body Injury** and **Vehicle Damage** child Case Types. The system resolves 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 the sample 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

©2025 Pegasystems Inc

Case, you can reuse your existing Case Types instead of creating a whole new business process. Reusing child Case Types increases the flexibility of your application, and helps you meet unique and dynamically changing needs. By using child Case Types as building blocks, you can resolve complex business cases and dependencies without planning and creating long, complicated Case Types.

Sharing data between Cases

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

The following figure shows a graph that illustrates sharing data between the parent and child Case Types:

©2025 Pegasystems Inc

Sharing data between Case Types

Reusing child Case Types in different scenarios

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

©2025 Pegasystems Inc

Sharing a child Case Type

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

What to do next: After you understand dependencies between your business processes, identify types of users that you need to involve in your Cases. For more information, see Identifying people involved in Case Types.

For relevant training materials, see the Creating a child Case module on Pega Academy.

Related concepts

- Case Types

Related tasks

- Creating a Case Type manually
- Creating a child Case Type in Case Designer

©2025 Pegasystems Inc

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

PEGA ©2025 Pegasystems Inc

tasks to users. After you define all of the actions in your business process, you can use Pega Platform to visualize, model, and automate your work. As a result, you can create an application that precisely reflects your business process. Pega Platform also offers tools that you can use to dynamically adjust actions to your changing requirements. For example, a user might perform an action only when it is relevant to a specific Case, such as a CSR who reviews a loan request only if a customer's income is higher than a specified amount.

How will users interact with my application?

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

The following figure shows a diagram that includes basic elements of an application to review job candidates. The diagram includes Personas involved in the process, main actions in the process, and interactive channels that Personas can use:

©2025 Pegasystems Inc

Main elements of a sample application

What to do next: After you define the crucial elements of your business processes, start modeling and visualizing your work. For more information, see Building Case Types.

Related concepts

©2025 Pegasystems Inc

- Defining Case participants
- Mobile features overview
- Conversational channels

Related tasks

- Notifying Case participants about Case events
- Enabling Case Processing in offline-enabled mobile apps

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 Lifecycle 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 Lifecycle. 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.

©2025 Pegasystems Inc

- 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 Lifecycle. 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:

©2025 Pegasystems Inc

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.

©2025 Pegasystems Inc

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 Collecting personal information and Collecting 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.

©2025 Pegasystems Inc

Steps

Steps are the smallest elements of a Case Lifecycle 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 Lifecycle. As a result, you can model Case Types that exactly meet your business requirements. For example, for a Collect personal information Process, you can add the following Steps:

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

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

Steps in a Process in Case Designer

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

©2025 Pegasystems Inc

Naming conventions

As a best practice, use descriptive, business-oriented names for Stages, Processes, and Steps, so that different stakeholders across your enterprise can understand the business value of each element. Typically, naming Stages, Processes, and Steps is within the responsibility of business representatives.

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 Case Lifecycle. 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 Lifecycle 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.

For relevant training materials, see the Defining a customer Microjourney module on Pega Academy.

Related concepts

- Case Lifecycle elements
- Stages in a Case Lifecycle
- Steps in a Case Lifecycle

Related tasks

- Processes in a Case Lifecycle

PEGA ©2025 Pegasystems Inc

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.

PEGA ©2025 Pegasystems Inc

The following figure shows sample scenarios in which you can reuse personal details of a customer across your application. For example, you can reuse the details on certain forms or for Cases related to filing a complaint or returning purchased objects:

Reusing data across application

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.

©2025 Pegasystems Inc

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 Lifecycle

To provide users with a series of logically ordered Steps towards their business goal, define a Case Lifecycle that orders actions into a sequence. For example, in an application for reviewing job candidates, you can define a sequence in which a hiring

PEGA ©2025 Pegasystems Inc

manager needs to approve a candidate before an HR worker starts an onboarding process for the candidate. Although with Pega Platform you can create flexible applications and respond to dynamically changing circumstances in an agile way, you can also define which actions are necessary to resolve a business process. As a result, you ensure that Case workers have enough data to resolve a Case in any scenario, and that your work meets all the regulations specific to your organization.

For more information, see Case Lifecycle elements.

Eliminating duplicate Cases

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

For more information, see Searching duplicate Cases.

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 Configuring Case-wide validation for correct data entry in fields and Validating input values on a form.

PEGA ©2025 Pegasystems Inc

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 Archiving and expunging Case data.

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.

PEGA ©2025 Pegasystems Inc

Related concepts

- Best practices for compliance and updatability

Related tasks

- 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.

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 Lifecycle, an application can autogenerate an email to the customer with the bank's decision about their loan, as in the following example:

©2025 Pegasystems Inc

A sample Case Lifecycle in Case Designer

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

For relevant training materials, see the Defining a customer Microjourney module and Designing a Case Lifecycle challenge on Pega Academy.

Blueprint for optimizing Case Type development process

As a best practice, use Pega GenAI Blueprint™ to develop the foundations of a new application. Describe your goals, and Pega GenAI Blueprint will instantly suggest optimal Case Type designs, based on GenAI and a vast repository of workflow best practices. By creating a Blueprint on https://pega.com/blueprint, and then importing the Blueprint to Pega Platform, you significantly accelerate the Case Type development process. The following figures show a sample use of Pega GenAI Blueprint to generate a Case Type in a car insurance application:

©2025 Pegasystems Inc

Generating a Case Lifecycle using Pega GenAI Blueprint

For more information, see Creating a new application from a Blueprint.

You can also create a Case Type manually or by using Pega GenAI Autopilot™. For more information, see Creating a Case Type manually and Creating a Case Type with Autopilot.

- **Case Types**
- **Case Lifecycle elements**
- **Extending a Data Model for a Case**
- **Designing the UI in Case Types**
- **Restricting user input in a field**

PEGA ©2025 Pegasystems Inc

Case Types

Save time and maximize efficiency when you implement similar business processes by defining Case Types. A Case Type is a visual representation of your business process, and a template for work that you can reuse for multiple instances of your business process. By creating a Case Type, you define the ultimate goal of a business process, the path that the Case must follow to a resolution, the people who are involved in processing the Case, and the data that the Case requires.

For example, in a Case Type for handling accident claims, you define actions such as collecting documents and reviewing vehicle damage, people such as a policyholder and an insurance company representative, and data such as a vehicle identification number and vehicle registration plate. When you create such reusable work templates, you save time and ensure that your business processes reach successful resolution.

Pega Platform™ refers to a main business process as a Parent Case Type, and to supporting business processes as child Case Types. A parent Case Type depends upon the results of the child Case Types that are together needed to reach the ultimate, expected outcome. To resolve a parent Case, you first need to resolve all of its child Cases. For example, a parent Case Type that represents an accident claim can only be resolved after child Case Types for vehicle damage and bodily injury are completed.

In this example, an insurance company representative creates an **Accident claim** Case, which is referred to as the parent Case, each time a policyholder reports a car accident. After your application verifies the policyholder's driving license and vehicle identification number, the system creates a **Vehicle damage** Case, which is the child Case that supports the **Accident claim** Case.

Based on the information from the policyholder, the insurance company representative then decides whether the application also needs a **Bodily injury** Case. After the claims adjusters process the **Vehicle damage** and **Bodily injury** Cases, the system automatically calculates the funds payable and updates the policy premium in the

PEGA ©2025 Pegasystems Inc

Accident claim Case. Agents who can approve the claim then receive a notification that the **Accident claim** Case is ready for review.

The following figure shows the sample business process with the two child Case Types:

A Parent Case Type that includes two Child Case Types in Case Designer

By reusing child Case Types in different scenarios, you save time, reduce costs and increase the flexibility of your application. For example, in the presented scenario, both vehicle and property insurance claims can include a bodily injury claim. By implementing the bodily injury claim as a separate Case Type, you can use the **Bodily injury** Case Type with both the vehicle and property insurance claims.

Naming conventions for Case Types

When you select a name for your Case Type, focus on the outcome of your business process rather than on the actions that users need to complete. The structure of Case Types is flexible and can adjust to changing business circumstances, when the goal remains unchanged. Ensure that you provide names that clearly convey the purpose of

©2025 Pegasystems Inc

your Case Type. For example, the name Financial operations is ambiguous and non-specific, but Loan application review, Mortgage request review, or Credit card dispute precisely communicate their purpose.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **Creating a Case Type with Autopilot**
- **Creating a Case Type manually**
- **Creating a Constellation Case Type in a traditional UI application**
- **Creating a child Case Type in Case Designer**
- **Creating a Case Type Rule form in Dev Studio**
- **Remote Case Types**
- **Creating Cases with a web embed**

Creating a Case Type with Autopilot

Improve efficiency and reduce workload by creating top-level Case Types with Pega GenAI Autopilot™. When you create a Case Type, you can select AI-suggested options instead of manually inputting Case workflow data, such as Case name, Stages, and Steps.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

PEGA ©2025 Pegasystems Inc

For more information, see Creating a new application from a Blueprint.

Before you begin:

Enable and configure generative AI in your application settings to use AI features. For more information, see Connecting to the generative AI provider in Pega Cloud.

Creating a Case Type

1. In the navigation pane of App Studio, click **Case Types**.
2. In the header of the **Case Types** section, click **New**.
3. In the **Case Type name** field, enter a name for the Case Type:
   - To enter a Case Type name manually, click the **Case Type name** field, and then enter a suitable name.
   - To select a Pega GenAI Autopilot suggested Case Type name, select a name from the list below the **Case Type name** field.

Autopilot suggested Case Type names

NOTE: AI-suggested Case Type names are based on the application name.

4. In the **Build Lifecycle and Data Model?** list, select **Use Autopilot to suggest a Lifecycle and Data Model**.

©2025 Pegasystems Inc

5. Click Next.

Result:

Autopilot creates and displays a Case Lifecycle based on the application and Case information provided.

An Autopilot generated Case Workflow

Editing the list of AI-generated Stages and Steps

6. **Optional:** To edit the details of the Autopilot generated Case workflow, in the **Add Stages and Steps** dialog box, define your new Case Type workflow:

Choices	Actions
Add a Stage	a. Click Add Stage to Case Type. b. In the Stage field, enter a name for the new Stage.
Add a Step	a. In the Process where you want to add a new Step, click Add Step to the Process.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **Step** field, enter a name for the new Step.
Edit a Stage or Step name	a. Click the Stage or Step that you want to edit. b. Delete the name and enter a new name for the Stage or Step.
Remove a Stage or Step	Next to the Stage or Step that you want to remove, click **Delete**.

7. **Optional:** To regenerate the AI-generated Case Workflow, click the menu icon and click **Regenerate**.
8. **Optional:** Regenerate the AI-generated Case Workflow with more context:
   a. Click the **Menu** icon.
   b. Click **Regenerate with more context**.
   c. In the **Provide additional context** field, enter additional information about your desired Case Workflow.
   d. Click **Regenerate**.
9. Click **Next**.

Generating and editing a list of AI-suggested fields

10. In the **Review AI-suggested fields** dialog box, click **Generate with AI**.

> **Result:**
>
> The Data Model generated with Autopilot for a Case Type automatically includes all the fields from the Case Type class hierarchy. Autopilot suggests only new fields that are not present in the system or inherited from the Case Type class.
>
> For more information, see Understanding class hierarchy and inheritance.

©2025 Pegasystems Inc

11. **Optional:** To edit the details of the AI-suggested fields, in the **Review AI-suggested fields** dialog box, define the fields according to your business needs:

Choices	Actions
Add a field	a. Click Add field. b. In the Field row, enter a name and ID. c. In the Type list, select a field type.
Edit a field name or ID	a. Click the Name or ID field that you want to edit. b. Delete the name and enter a new name for the Name or ID.
Edit a field type	a. Click the Type field that you want to edit. b. Select a Type from the list.
View and edit a data object	a. Click View folder contents. b. **Optional:** Edit the data object name, record configuration, or fields. c. Click Submit. **NOTE:** View folder contents is only available for Embedded Data and Data reference Field Types.
Remove a field	Click the Delete icon to the right of a field to remove it.

12. Click Submit.

©2025 Pegasystems Inc

For example:

The following video shows creating a Case Type by using Pega GenAI Autopilot to suggest the workflow and Data Model:

What to do next:

- Create child Case Types that represent dependencies that you resolve before the top-level Case Type can reach the resolution. For more information, see Creating a child Case Type in Case Designer.
- Add automated Steps to your Processes to improve workflow efficiency. For more information, see Adding an automated Step to a Process.

Related concepts

- Fields as building blocks for data objects

Creating a Case Type manually

Improve work processing in your application by creating 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 Case Type that represents a hiring process, and then reuse the Case Type every time when you want to process a job application from a new candidate.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on Pega GenAI™ and a vast repository of workflow best practices. For more information, see Creating a new application from a Blueprint.

©2025 Pegasystems Inc

1. In the navigation pane of App Studio, click **Case Types**.
2. In the header of the **Case Types** section, click **New**.
3. In the **Case Type name** field, enter a name for the Case Type.

> **NOTE:** If you configure your Case to enable Generative AI, you can select AI suggested Case Type names. For more information, see Integrating with Pega GenAI.

4. Expand the **Advanced** node, and then, in the **Type** field, select what kind of a Case Type you want to create:
   - To add a standard Case Type with traditional UI to your application, select **Standard**.
   - To add a questionnaire Case Type to your application, select **Questionnaire**. For more information, see Creating a questionnaire.
   - To add a Remote Case Type to your application, select **Remote**. For more information, see Creating Remote Case Types.
   - To add a Constellation Case Type to your application, select **Standard (Using Constellation technology)**. For more information, see Creating a Constellation Case Type in a traditional UI application.
5. In the **Reuse assets from** list, select the Case Type from which you want to source assets.
6. **Optional:** To add data to your Case Type from a spreadsheet, select the **Create Data Model using spreadsheet**.
7. Click **Next**.
8. If you chose to create a Data Model for your Case Type from a spreadsheet, in the **Select file** window, complete the data import process.
   For more information, see Creating a Data Model from a spreadsheet.

©2025 Pegasystems Inc

Result:

Your new Case Type includes a default **Create** Stage with a Create Process, as shown at the end of the procedure example in the first figure. The second figure shows the same Case Type already populated with Stages and Processes.

Creating a Case Type

Populated Case Type

©2025 Pegasystems Inc

What to do next:

- Divide work into phases by adding Stages to your Case Type. For more information, 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. For more information, see Creating a child Case Type in Case Designer.
- Create a UI for a Constellation Case, see Creating Views, Configuring a Full Page View, and Configuring Views.
- Create a UI for your traditional UI Case. For more information, see Summary panel Views, Preview panel, and Populating Views.

- The Create Stage
- Case Types

Creating a Constellation Case Type in a traditional UI application

Include Constellation Cases in applications that use the traditional UI architecture. This ability to blend the two Pega Platform™ architectures enables you to integrate Constellation into your existing product while maintaining application continuity.

Before you begin:

- Ensure that your application supports Constellation components. For more information, see Constellation in traditional UI.
- Familiarize yourself with limitations of mixed architectures. For more information, see Limitations when using Constellation with traditional UI.

PEGA ©2025 Pegasystems Inc

> **NOTE:** In a Constellation application, the *PegaAPI* role is added automatically to your role list when you create a new application. For traditional applications using Constellation Cases, you must add the *PegaAPI* role manually to your access configuration to ensure proper operation of functionalities such as mashup.

1. In the navigation pane of App Studio, click **Case Types**.
2. In the header of the **Case Types** section, click **New**.
3. In the **Case Type name** field, enter a name for the case type.
4. Expand the **Advanced** node, and then select the **Create with Constellation architecture** checkbox.
5. **Optional:** To define additional Case settings, in the **Advanced** node perform the following actions:
   a. In the **Type** list, select **Standard (Using Constellation technology)**.
   b. To reuse assets from another Case Type, in the **Reuse assets from** list, select a Case Type.

      > **NOTE:** Building Constellation Cases based on assets from traditional UI is not supported. Reusing assets from other Constellation Cases is supported.

   c. To add data to your case from a spreadsheet, select the **Create Data Model using spreadsheet**.
6. Click **Next**.
7. If you chose to create a Data Model for your case from a spreadsheet, in the **Select file** window, complete the data import process.
   For more information, see Creating a Data Model from a spreadsheet.

©2025 Pegasystems Inc

What to do next:

- Divide work into phases by adding Stages to your Case Type. For more information, 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. For more information, see Creating a child Case Type in Case Designer.
- Create a UI for a Constellation Case, see Creating Views, Configuring a Full Page View, and Configuring Views.

- **Reusing a Constellation Case Type from an existing application**
- **Limitations when using Constellation with traditional UI**

Reusing a Constellation Case Type from an existing application

Take advantage of Constellation Case Types by adding them to applications that use traditional UI. You can add existing Constellation Cases to both UI Kit and Theme Cosmos applications, and maintain application continuity while you rebuild your UI.

Before you begin:

- Identify the Constellation application whose Case Types you want to reuse.

  In this scenario, the source application name is **ConstellationApp**, and the Case Type name is **Constellation Case**.

- Learn about the limitations of using a mixed UI. For more information, see Limitations when using Constellation with traditional UI.

Adding the Constellation Case Type to the application definition

©2025 Pegasystems Inc

1. In the header of Dev Studio, click the name of the application, and then click Definition.
2. On the Definition tab, in the **Built on applications** section, add the Constellation application that hosts the Case Type that you want to add.

For example:

Add ConstellationApp. For more information, see Adding built-on applications.

The Built on applications section on the Definition *tab of the application definition*

3. On the Cases & data tab, make the Case Type available to application users by adding it to the list in the **Case Types** section.

For example:

Add Constellation Case. For more information, see Managing application Cases and data.

©2025 Pegasystems Inc

The Constellation *Case on the* Cases & data *tab of the application definition*

Result:

The Constellation Case is now available in your application.

What to do next:

Add the Case to the create menu of your traditional application. For more information, see Organizing the main navigation for a standard work Portal.

Limitations when using Constellation with traditional UI

Pega Platform™ '24.2 supports the ability to include Constellation Cases in your application. While this new paradigm enables you to expand your existing traditional application with elements of the Constellation UI, it remains an evolving feature and includes some limitations.

As of Pega Platform '24.2, the following features are not supported when using Constellation Cases in traditional applications:

- Seamless user experience.

  Using two architectures for your Cases might result in inconsistent user experience and navigation patterns. In terms of accessibility, the two technologies use different navigation patterns.

©2025 Pegasystems Inc

- Self-service.

  Using Constellation Cases in traditional UI is only supported for back office uses. Features such as traditional DX API, starter packs or mashups are not supported for Constellation Cases.

- Cross referencing between Cases that use different UIs.

  For example, navigating and accessing traditional UI Cases from a Constellation Case Page (or vice versa) is not supported.

- Get next work on Case Pages.

  Get next work is still supported on Landing Pages.

- Data and Case references defined in the Constellation Case.

  The links associated with those fields do not work at run time.

- Theme sharing between technologies.

  Themes must be defined separately for the Constellation UI and the traditional UI.

- Custom Actions in Preview links.

  In Theme Cosmos applications, Preview links with action set configurations that derive the Case key on the fly are not supported.

- Constellation Cases in custom Portals that were created by using AJAX containers.

  Portals configured on the **Channels and interfaces** Landing Page are supported.

- Case reference Data Pages that do not have an API status.

  Case references in Constellation require the Data Page to have an API status. To ensure that your Case reference works correctly, you must set the Data Page status to API.

PEGA ©2025 Pegasystems Inc

- Building Constellation Cases based on assets from traditional UI.

  Reusing assets from other Constellation Cases is supported.

- Opening Constellation Cases from mobile apps or mobile browsers.
- Localization.

  Labels and messages from Constellation Case Types created in the traditional application are not localized.

- Preview Actions on links that use the *OpenWorkItem* Action form a built on app are not supported.

Creating a child Case Type in Case Designer

Create child Case Types to ensure that users of your application address all elements in a complex business process before resolving the Case. When you supplement your business processes with child Case Types, you create dependencies that these Case Types represent.

For example, you can create a Case Type that represents a review of an accident claim, and then supplement the Case Type with child Case Types that visualize the processes of estimating vehicle damage and injuries. To resolve the **Accident claim** Case, a worker first resolves the **Vehicle damage** and **Injury** Cases. Different workers can handle each child Case in parallel to resolve Cases more efficiently.

> **NOTE:** If you want to add a child Case Type that you created in a traditional UI application to a Case Type in a Constellation application, see Adding a traditional Case as a child Case to Constellation and Limitations when using traditional UI with Constellation.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.

©2025 Pegasystems Inc

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

For example:

The following figure shows a sample configuration for transferring information about an accident insurance claim from the parent Case Type to the child Case Type:

©2025 Pegasystems Inc

Transferring data to a child Case Type

7. Click **Save**.

Result:

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

- **Creating child Case Types in Dev Studio**

- Case Types
- Understanding Case hierarchy
- Building Case Types
- Enabling automatic creation of child Cases
- Sharing data between parent and child Cases
- Creating contextual Cases in traditional UI applications
- Starting child and Remote Cases from the main Case context

©2025 Pegasystems Inc

Creating child Case Types in Dev Studio

Define a more structured work processing in your organization by creating child Case Types. When you create a child Case Type, you define a set of actions that are dependent on the main business process, and that users need to complete before the parent Case Type reaches a resolution. As a result, you avoid creating long and complex Case Types and you deliver more granular and flexible applications that are easier to maintain

For example, the process of hiring an employee for a managerial position might require more complex and longer background check. By defining the background check as a child Case Type, you have the possibility to reuse this business scenario in only the relevant circumstances. A Case of hiring a manager can reach a resolution only after the child Case of running a background check completes.

Before you begin:

Create a parent Case Type. For more information, see Creating a Case Type manually.

NOTE: Creating Case Type hierarchy in Dev Studio is suitable for advanced developers. To take advantage of low-code tools, create child Case Types in App Studio. For more information, see Creating a child Case Type in Case Designer.

1. In the navigation pane of Dev Studio, click **Case Types**.
2. Hover over the name of a Case Type, and then click **More** > **Add a child Case Type**.
3. In the **Add Case Type** dialog box, define the Case Type that you want to use:
   - To create a new Case Type, select **New Case Type**, and then, in the **Name** field, enter a label for your Case Type.

©2025 Pegasystems Inc

- To reuse an existing Case Type, select **Existing Case Type**, and then, in the **Select an existing Case Type to add as a child** list, select a Case Type that you want to use.

4. **Optional:** If you create a new Case Type, to define resources that your Case Type can access by using class inheritance, expand the **Advanced Settings** section:
   - To change the parent class from which your Case Type directly inherits resources, in the **Derives from (Directed)** field, enter a new class.
   - To change the class from which your Case Type inherits resources by using the class name structure, in the **Derives from (Pattern)** list, select a new class.
   - To change the Ruleset or the Ruleset version to store your new case type, in the **Ruleset** and **Ruleset version** lists, select the new values.
5. Click **Submit**.

> **For example:**
>
> The following video shows a user adding a new child Case Type to a Loan Approval Case Type:

> **What to do next:**
>
> Populate your Case Types with actions that advance your business processes closer towards a resolution. For more information, see Case Lifecycle elements.

- Understanding Case hierarchy
- Identifying Case Type elements
- Case Types
- Understanding class hierarchy and inheritance
- Adding a traditional Case as a child Case to Constellation

PEGA ©2025 Pegasystems Inc

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 Lifecycle Designer tool, which helps you visualize your business process in a low-code way. For more information, see Creating a Case Type manually.

1. In the header of Dev Studio, click **Create menu** > **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.
5. Click **Create and open**.

©2025 Pegasystems Inc

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 the Case.
10. **Optional:** To enable the navigation button to go back to a previous Step in a Process, select the **Allow users to go back to the previous Step** checkbox.
11. **Optional:** To add more supporting Processes, click the **Add a row** icon, and then provide another Process.
12. **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.
13. 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.

PEGA ©2025 Pegasystems Inc

14. **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.
15. Click **Save**.

- Case Types
- Introduction to Case Management
- Adding supporting optional Processes to Cases
- Searching duplicate Cases

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.

©2025 Pegasystems Inc

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 using a value that a user provides for a phone number in a `Review job candidate` parent Case.

   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** checkbox, and then, in the text field, enter a Data Transform.

©2025 Pegasystems Inc

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** checkbox. 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 Run a background check child Case Type only if a job candidate applies for a manager position.
Start a child Case automatically	a. Select the **Automatic instantiation** checkbox. 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

©2025 Pegasystems Inc

Choices	Actions
	the **Permitted When** field, enter a When Rule that defines when the child Case starts. **For example:** Select a When Rule that starts a `Run a background check` 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. **Optional:** To indicate that the parent Case Type cannot reach resolution before the child Case Type is complete, select the **Required** checkbox.
8. **Optional:** To add more child Case Types, click **Add a row**, and then repeat steps 4 through 7.
9. Click **Save**.

Related concepts

©2025 Pegasystems Inc

- Understanding Case hierarchy
- Case Types

Related tasks

- Creating a child Case Type in Case Designer

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.

©2025 Pegasystems Inc

> **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

- Configuring Case-wide validation for correct data entry in fields
- Referencing properties

©2025 Pegasystems Inc

- Building logic and calculating values in your application

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. You can use additional processing to switch the Case to an Alternate Stage without canceling the current assignment, which helps you to track Service-Level Agreement (SLA) requirements.

Adding Stages through a Rule form is suitable for advanced developers. If your use Case does not require advanced configurations, populate your Case Types by using the Case Designer tool, which helps you visualize your business processes in a low-code way. For more information, see Stages in a Case Lifecycle.

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 the text field, enter a Stage name.

©2025 Pegasystems Inc

5. Configure conditions for skipping a Stage:
   - To ensure that a Case always enters a Stage, in the **Skip Stage** list, select **Never**.
   - To skip a Stage by evaluating a When Rule, in the **Skip Stage** list, select **When Rule**, and then in the text field, enter a When Rule name.
   - To skip a Stage by using an Expression, in the **Skip Stage** list, select **Expression**, and then in the text field, enter an Expression name.

> **NOTE:** You can configure skip conditions for Primary Stages only. A Case cannot skip the last Stage.

6. Configure Case behavior when a Stage is complete:

Choices	Actions
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.

©2025 Pegasystems Inc

Choices	Actions
	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** checkbox. d. **Optional:** To delete any dependent Cases after the Case reaches resolution, select the **Delete open Assignments** checkbox.

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.

©2025 Pegasystems Inc

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-Level 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.

©2025 Pegasystems Inc

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

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.

©2025 Pegasystems Inc

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 CSR can update the details without interrupting the main Case processing.

12. **Optional:** To enable the navigation button to go back to a previous Step in a Process, in the **Case-wide supporting Processes** section, select the **Allow users to go back to the previous Step** checkbox.
13. **Optional:** To enable changing to an Alternate Stage without canceling the current Assignment, in the **Check for additional processing required before Assignment is ready** section, configure the condition and the Stage that you want to launch after the condition is met.
14. Click **Save**.

- Building Case Types
- Creating a Primary Stage
- Creating an Alternate Stage
- Processes in a Case Lifecycle
- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools
- 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.

PEGA ©2025 Pegasystems Inc

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 checkbox in the **Required for resolution** column.
7. **Optional:** To change the name of the property, in the **Property reference** column, click **Rename**, and then, in the text box, enter a new name.
8. **Optional:** To add more attachment categories, repeat steps 3 through 7.
9. Click **Save**.

©2025 Pegasystems Inc

For example:

The following figure shows an attachment category for photos that are required in a Case Type to review vehicle damage insurance claims:

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

Attachment category configuration

- Securing Case access
- Controlling access to Case attachments
- Referencing properties

Remote Case Types

In organizations that use multiple Pega Platform™ applications, Remote Case Types enable users to create and reference Cases across applications without switching logins. While task processing still occurs in the originating (producer) application, Remote Case Types streamline Case creation and visibility into a single application context.

NOTE: Remote Case Types enable a Pega Platform application (consumer) to create and reference Cases in another Pega Platform application (producer). However, task processing - such as submitting Assignments or completing Steps - must still be performed in the producer application. Remote Case Types do not support cross-application task processing.

IMPORTANT: Federated Case Management (FCM) has evolved into the Remote Case Types functionality in Pega Infinity™. FCM has moved away from its underlying technical dependency on the IAC Gateway, which was

©2025 Pegasystems Inc

> deprecated in Pega Infinity 8.1. This means that FCM is still available as the Remote Case Types feature, however, it does not support integration with non-Pega applications.

System architecture

For Remote Cases to work, a producer system hosts an application that has a Case Type that you want to process at run time. Meanwhile, a consumer system hosts an application that you want to connect to the Case Type. In addition, you can configure one system as both a producer and a consumer if you host multiple applications in one system. First, you create a Case Type in the producer system, and then create this Case Type as the Remote Case Type in the consumer system. You populate your Case Types with Actions and Data Model fields in the producer system, as the consumer system only references the Case Types. You cannot edit the Case Types in the consumer system. The following figure presents a sample use Case that shows how Remote Case Types work:

A sample model of Remote Case Types

In the sample use Case, in the VIP financial operations application, CSRs can process Case Types that come from other applications:

©2025 Pegasystems Inc

- Loan requests
- Mortgage requests
- Credit card operations

Implementation considerations

Consider the following factors before you implement Remote Case Types:

- Build both producer and consumer applications by using Constellation. Only applications that are based on Constellation support Remote Case Types.
- Before you can work on a Remote Case Type in a Remote application, establish a secure connection between the producer and consumer systems.
- Ensure that you configure both producer and consumer systems by using the same operator ID.
- Ensure that you configure both producer and consumer systems to work with SSL certificates.
- If your consumer system is in Pega Platform version 8.8, ensure that the producer system is also in Pega Platform version 8.8 so that the remote application from the consumer system can communicate with your producer system.
- After an update to a later release of Pega Platform, you must override the **Remotecasetype_Consumer** token profile again. For more information, see Configuring a consumer system for Remote Case Types.
- After you import Remote Case Types to a different environment, you must edit the connection details of each of your Remote Case Types.

Creating Remote Cases

You can configure your Remote Cases to be available at run time in the following ways:

In the navigation pane of the Web Portal

Users click **Create** in the navigation pane of the Web Portal, and then select the Remote Case Type to create a new Remote Case.

©2025 Pegasystems Inc

For more information, see Creating Cases from the navigation pane of a Portal, Organizing the main navigation for a standard work Portal, and Organizing the main navigation for a self-service Portal.

In the Create Case Step in the Lifecycle of the current Case

The system starts a Remote Case when the current Case reaches the **Create Case** Step.

For more information, see Starting child and Remote Cases from the main Case context.

As an optional Action in the Actions menu of the current Case

Users select a relevant Action that starts a Remote Case from the **Actions** menu of the current Case.

For more information, see Creating Cases from the Actions menu of another Case.

- **Configuring a consumer system for Remote Case Types**
- **Configuring a producer system for Remote Case Types**
- **Creating Remote Case Types**
- **Updating connection details for Remote Case Types**
- **Fetching a list of selected Remote Cases with a queryable Data Page**
- **Error messages for Remote Case Types**

- Constellation UI
- Creating an operator ID
- Starting child and Remote Cases from the main Case context

PEGA ©2025 Pegasystems Inc

Configuring a consumer system for Remote Case Types

To process Cases that come from another application, configure the system that hosts your destination application as a consumer. As a result, at run time, users can perform work from multiple applications within the context of one consumer application, without the need to switch between different products.

For example, a customer service representative can use a Financial operations application to work on Cases that belong to the Loan requests and Mortgage requests applications. In this case, the Financial operations application is on a consumer system, while the Loan requests and Mortgage requests applications operate on a producer system.

Before you begin:

- Ensure that the application for processing Remote Case Types at run time uses Constellation. For more information, see Constellation UI.
- Secure the connection between producer and consumer systems. For more information, see Configuring an OAuth 2.0 authentication profile and Creating a generation JSON Web Token profile.
- Prepare `.jks` files of the consumer system to upload to the keystore and truststore. For more information, see Creating keystore.jks and truststore.jks files.
- Ensure that you configure the consumer system with the same operator ID that the producer system uses.
- Ensure there is at least one unlocked Ruleset open in the consumer system to register the Remote Case Type or to update connection details. For more information, see Branches and unlocked Rulesets.

Configuring a consumer system requires a keystore and truststore that support connection to the producer system, as well as copying and then editing a token profile that Pega Platform includes by default.

PEGA ©2025 Pegasystems Inc

1. Create a truststore with the `Remotecasetype_Truststore` name, and then upload the `.jks` truststore file of the producer system.
   For more information, see Creating a keystore for application data encryption.
2. Create a keystore with the `Remotecasetype_Keystore` name, and then upload the `.jks` keystore file of the consumer system.
   For more information, see Creating a keystore for application data encryption.
3. In the navigation pane of Dev Studio, click **Records**.
4. Expand the **Security** category, and then click **Token Profile**.
5. In the list of instances, open the **Remotecasetype_Consumer** token profile.

> **NOTE:**
> - For remote Case Types to work correctly, configure the token profile that Pega Platform™ includes by default instead of copying or creating a new instance of the token profile.
> - After an update to a higher release of Pega Platform, you must override the **Remotecasetype_Consumer** token profile again.

6. In the **Security** section of the token profile, configure security settings:
   a. In the **Security configuration** list, select **Signature**.
   b. In the **Signature type** list, select **Asymmetric**.
   c. In the **Signature type algorithm** field, press the Down arrow key, and then select **RS256**.
   d. In the **Keystore** field, enter the keystore that you created in step 2.

> **TIP:** The JSON web key set URI from the **Security** section of this token profile is the URL that you use when you create a truststore for the producer system.

©2025 Pegasystems Inc

Result:

When you select the keystore, the system autopopulates the alias value.

e. In the **Password** field, enter the password related to the keystore.

7. Click **Save**.

For example:

The following figure shows security configuration for the token profile of the consumer system:

Security section of a token profile

©2025 Pegasystems Inc

What to do next:

Create a connection between your applications so that you can publish Remote Case Types. For more information, see Creating Remote Case Types.

- Key management service for data encryption
- Authentication

Configuring a producer system for Remote Case Types

To pass Remote Case Types to another application, configure the system that stores your Case Types as a producer system. As a result, you provide a framework for Case workers to process work from multiple applications without changing context, which promotes efficiency and saves time in your organization.

For example, your system might host Loan requests and Credit card operations applications. You want to publish Case Types from these applications to a VIP customers operations application so that users can perform work by logging in to this application only. You configure the system with the Loan requests and Credit card operations applications as the producer system first.

Before you begin:

- Ensure that your applications in the producer system use Constellation. For more information, see Constellation UI.
- Secure the connection between producer and consumer systems. For more information, see Configuring an OAuth 2.0 authentication profile and Creating a generation JSON Web Token profile.

Accessing Remote Case assets that are available in the producer system requires performing security configurations. Pega Platform provides the OAuth 2.0 client registration, identity mapping, and token profile that you copy into your Ruleset and

PEGA ©2025 Pegasystems Inc

edit to meet your business needs. The security configuration also requires your own truststore.

- Key management service for data encryption
- Constellation UI
- Authentication

Preparing a token profile for remote Case Types

Prepare a token profile to secure data exchange between two parties, such as producer and consumer systems.

Before you begin:

Create a truststore that references the URL that is the JSON web key set URI in the `Remotecasetype_Consumer` token profile, as in the following example:

©2025 Pegasystems Inc

Creating a truststore with the URL from a token profile

For more information, see Creating a keystore for application data encryption and Configuring a consumer system for Remote Case Types.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **Token Profile**.
3. In the list of instances, open the **Remotecasetype_Producer** token profile.
4. In the form header, click **Save** > **Save as**.
5. On the **Save Token Profile As** form, in the **Name** field, provide a unique label, and then click **Create and open**.

©2025 Pegasystems Inc

For example:

The following figure shows a sample new label for a token profile:

Copying a token profile

NOTE: Make note of the token profile name because this name is required when preparing the identity mapping.

6. In the form header, in the **RS** section, click **Edit**, and then, in the text field, provide the Ruleset to associate with your application.
7. In the **Claims mapping** section, click **Add a row**.
8. In the **Claim name** field, enter `sub`, and then, in the **Property** field, enter `pxRequestor.pyUserIdentifier`.
9. In the **Claims mapping** section, click **Add a row**.
10. In the **Claim name** field, enter `exp`, and then, in the **Property** field, enter `OperatorID.pyExpirationDate`.

©2025 Pegasystems Inc

Result:

The following figure shows a correct configuration of the **Claims mapping** section:

Claims mapping for the producer system

11. In the **Security** section, in the **Truststore** field, enter the truststore that you created for the producer system.

Result:

The following figure shows a correct security settings configuration:

Security settings for the producer system

©2025 Pegasystems Inc

12. Click **Save**.

- Security operations
- Creating token profiles
- Copying a Rule or data instance

Preparing identity mapping for remote Case Types

After you prepare the token profile for your producer system, prepare identity mapping to ensure that Pega server correctly identifies operators in your application. You also define how to map the user identity information for use in the Pega application.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **Identity Mapping**.
3. In the list of instances, open the **Remotecasetype_Producer** identity mapping.
4. In the form header, click **Save** > **Save as**.
5. On the **Save Identity Mapping As** form, in the **Name** field, enter a unique label, and then click **Create and open**.
6. In the form header, in the **RS** section, click **Edit**, and then, in the text field, provide the Ruleset to associate with the identity mapping.
7. In the **Token Validation** section, in the **Token processing profile** field, enter the name of the token profile that you created for the producer system, as in the following example:

   For example:

©2025 Pegasystems Inc

Token profile for identity mapping

8. Click **Save**.

- Authentication
- Authenticating requests in services
- Creating an identity mapping data instance
- Copying a Rule or data instance

Preparing OAuth 2.0 client registration for Remote Case Types

To ensure that remote applications from consumer systems can properly communicate with your producer system, prepare the OAuth 2.0 client registration in your producer system. As a result, you create a secure and reliable connection between your producer and consumer systems.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **OAuth 2.0 Client Registration**.
3. In the list of instances, open the **Remotecasetype_Producer** OAuth 2.0 client registration.
4. In the form header, click **Save** > **Save as**.

©2025 Pegasystems Inc

5. On the **Save OAuth 2.0 Client Registration As** form, in the Client Name field, enter a unique label, and then click Create and open.
6. In the form header, in the **RS** section, click Edit, and then, in the text field, provide the Ruleset to associate with your application.

For example:

Associating a Rule with a Ruleset

7. In the **Client Credentials** section, download the client ID and client secret by clicking View & download.

> **NOTE:** You provide the client ID and client secret credentials when you register a remote system during creation of a Remote Case Type. Ensure that you store the credentials safely.

8. In the **Supported grant types** section, select JWT Bearer, and then, in the Identity mapping field, enter the identity mapping that you created for the producer system.
   For more information, see Preparing identity mapping for remote Case Types.
9. Click Save.

- Authentication
- Creating an authentication profile
- Copying a Rule or data instance

©2025 Pegasystems Inc

Creating Remote Case Types

Provide tools for fast and seamless work processing for organizations that use multiple applications by creating Remote Case Types. With Remote Case Types, users can process work from one application within another application, without changing the context and logging in to many different products.

For example, a customer service representative (CSR) who processes loan requests and mortgage requests can process work by using only one application, even if mortgage and loan requests come from two different applications. To provide this option for a CSR, in a Loan operations application, you create a Loan request Case Type, and then associate the Case Type with the Mortgage operations remote application. As a result, the CSR can create and process loan request Cases from the Mortgage operations application.

Before you begin:

- Create a connection between the application that includes the Case Type and the application that you want to use to process the Cases at run time. For more information, see Configuring a consumer system for Remote Case Types and Configuring a producer system for Remote Case Types.

  ⚠ **IMPORTANT:** The consumer and producer applications must use Constellation. For more information, see Constellation UI.

- In the producer system, create the Case Type that you want to define as a Remote Case Type in the consumer system, and then populate the Case Type with data objects and actions. For more information, see Case Types and Case Lifecycle elements.
- Log in to the consumer system.

PEGA ©2025 Pegasystems Inc

NOTE: When you create a Remote Case Type, the system creates two Data Pages that fetch a list and the IDs of Cases of this Case Type. For example, when you create a `Review loan request` Case Type, the system creates the `D_ReviewLoanRequestList` and `D_ReviewLoanRequest` Data Pages.

IMPORTANT: Do not delete these Data Pages in the producer system.

You can also use Pega Process Fabric Hub™ to manage your remote Case Types. Pega Process Fabric Hub consolidates work from multiple applications, greatly enhancing the way how your enterprise can manage and process Cases from multiple distributed applications within a single interface. For more information, see Pega Process Fabric Hub .

1. In the navigation pane of App Studio, click **Case Types**.
2. In the **Case Types** header, click **New**.
3. In the **Create Case Type** dialog box, enter the Case Type details:
   a. In the **Case Type name** field, enter a label for your Case Type, for example, `Loan requests`.
   b. Expand the **Advanced** section, and then, in the **Type** list, select **Remote**.
   c. Click **Next**.
4. Connect the Case Type to a remote application:

Choices	Actions
Use an existing connection	In the **Connect to remote application** dialog box, in the **Remote application** list, select the application that you want to connect to the Case Type, and then click **Connect**.

©2025 Pegasystems Inc

Choices	Actions
Create a new connection with a remote application	a. In the **Connect to remote application** dialog box, in the Remote application list, select Create new remote application. b. In the Name field, enter the name of your remote application, for example, `Mortgage operations`. **NOTE:** This application already exists in your producer system. You create a connection between your current application and the producer application. c. In the System URL field, enter the URL of the producer system. For multitenant systems, ensure that you include the tenant hash in the URL. For other systems, ensure that you include the context root in the URL. **For example:** For multitenant system, a sample URL is `https://DNS/prweb/app/`*{app alias}*`/`*tenantHash*`/` !

©2025 Pegasystems Inc

Choices	Actions
	STANDARD. For other systems, a sample URL is `https://domain/`*context-root*`/auth-servlet-path`. d. In the **Client ID** and **Client secret** fields, enter the credentials for connecting the Case Type to the application. You generate the credentials when you configure the producer system. **For example:** The following figure shows a sample configuration for creating a connection with a remote Mortgage operations application: *Connecting to a remote application* e. Click **Connect**.
Connect to the Pega Process Fabric Hub	For more information, see Creating Cases on traditional applications from Constellation applications.

©2025 Pegasystems Inc

5. In the **Select Case Type** dialog box, select a Case Type to source from the producer system:
   a. In the Application list, select the application that stores the Case Type.
      If your producer system has only one application that is configured to support Remote Case Types, the system autopopulates the Application field. The list displays only those applications that you can access in the producer system.
   b. In the Case Type list, select the Case Type that you want to source from the producer application.
   c. Click Next.
6. **Optional:** If you create a new connection with a remote application, to override default configuration for your application in other environments, such as staging production, provide new connection details:
   a. In the **Environment configuration** dialog box, select the Edit connection details in other environments checkbox.
   b. Provide connection details as necessary.
   c. Click Next.
7. Import the Data Model from the producer system:
   a. On the Data Model tab of the Case Type from the producer system, click Add fields.
   b. In the **Add fields** dialog box, select the fields from the Case Type Data Model that you want to import to process in the consumer system.
   c. Click Submit.
8. Click Save.

Result:

At run time, your consumer application can process Cases of the selected Case Type from the producer application.

©2025 Pegasystems Inc

What to do next:

Configure the way in which Case workers interact with Remote Case Types in the consumer application at run time. For more information, see Creating Remote Cases.

- Starting child and Remote Cases from the main Case context
- Identifying Case Type elements
- Building Case Types
- Extending a Data Model for a Case

Updating connection details for Remote Case Types

To ensure that the connection between your consumer and producer applications works correctly after you make changes in your producer system, update the connection details in the consumer system.

For example, if you need to change the URL of your producer system, reflect that change in the connection configuration. You can change the system URL, client ID, and client secret.

Before you begin:

- Ensure there is at least one unlocked Ruleset open in the consumer system to register the Remote Case Type or to update connection details. For more information, see Branches and unlocked Rulesets.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Connection** tab, in the **Connection details** section, click **Edit connection**.
3. In the **Edit connection** dialog box, edit the details as required, and then click **Submit**.

©2025 Pegasystems Inc

4. Click **Save**.

> **What to do next:**
>
> Configure the way in which Case workers interact with Remote Case Types in the consumer application at run time. For more information, see Creating Remote Cases.

- Key management service for data encryption
- Constellation UI
- Authentication

Fetching a list of selected Remote Cases with a queryable Data Page

If your business scenario requires fetching only selected records from the remote application, create a query field and pass your own parameters to the query so that you get only the Cases that meet specific criteria, such as the Case creator's ID or a Case ID.

When you create a Remote Case Type, the system automatically creates two Data Pages that fetch a list and the pyIDs of Cases of this Case Type. For example, when you create a **Review Loan Request** Case Type, the system creates the *D_ReviewLoanRequestList* and *D_ReviewLoanRequest* Data Pages. When you use the Data Pages in a query field to fetch Remote Cases data, the query gets a list of all the Remote Cases that are created for a given Case Type. To get a list of selected Cases based on your criteria, you create a new Data Page with your filter logic.

In this scenario, you create a query field that calls a Data Page in the remote Pega application to fetch a list of selected records from the remote system. The Data Page uses a Data Transform and an Activity Rule to pass your filter logic.

The Data Transform uses Pega Digital Experience (DX) API to communicate with the server when creating Remote Cases. For more information, see Data endpoints .

©2025 Pegasystems Inc

Creating a Data Transform with your filter logic:

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Data Model** category, and then click **Data Transform**.
3. In the list of Data Transforms, open and override the *pySampleRequestForFetchingRemoteCases* Data Transform:
   a. Click **Save as**.
   b. In the **Label** field, enter the title or description of the Data Transform.
   c. **Optional:** To edit the name, in the **Identifier** field, click **Edit**, and then enter a new name.
      By default, the **Identifier** field inherits the name from the **Label** field. A valid identifier name starts with a letter and consists of letters, digits, and underscores.
   d. In the **Apply to** field, select a class of the Remote Case Type class.
   e. In the **Add to Ruleset** list, select the Ruleset in which you want to include the Data Transform.
   f. **Optional:** To associate the Data Transform with a work item, in the **Work item to associate** field, press the Down arrow key, and then select a work item.
   g. Click **Create and open**.
   h. On the **Definition** tab, define the Remote Case Type fields and filter logic for the Data Transform request.

      For example:

      The following sample Data Transform is configured to filter Remote Cases based on the ID of the user that created a Remote Case:

©2025 Pegasystems Inc

A sample Data Transform for filtering Remote Cases based on operator ID

NOTE: If you want to display the Remote Case as a link, ensure that you define the **pzInsKey** and **pyID** fields on the *RequestParamPage.query* page on the **Pages & classes** tab of the Data Transform.

i. On the **Pages & classes** tab, set the *RequestParamPage* page with the filters and query fields that you want to pass to the *data_views/{data_view_ID}*DX APIrequest.
For more information, see Data endpoints and POST /data_views/{data_view_ID} .

j. Click **Save**.

Using the Data Transform as a parameter in an out-of-the-box wrapper Activity

4. Expand the **Technical** category, and then click **Activity**.
5. In the list of Activities, open and override the *pyFetchRemoteCaseInstancesWrapper* Activity:
   a. On the **Parameters** tab, in the **Parameters** section, in the **DataTransform** field, enter the name of the Data Transform from step 3.

©2025 Pegasystems Inc

b. In the **RemoteClass** field, enter the name of the Remote class List-structure Data Page in the consumer system.
c. In the **ListDataPage** field, enter the name of the Remote class List-structure Data Page of the producer system.
d. Click **Save**.

Using the Activity in a Data Page for fetching Remote Case instances based on the filters specified in the Data Transform:

6. Create a Data Page in the Remote Case class of the consumer system.
   a. In the header of Dev Studio, click **Create** > **Data Model** > **Data Page**.
   b. On the **Create Data Page** page, in the **Data Page Record Configuration** section, in the **Label** field, enter a name for the Data Page.
   Data Pages have names that start with `D_` and are case-sensitive. Use only letters, digits, and underscores. For more information, see Page names and reserved pages.

   > **For example:**
   >
   > Enter *Case Type Name*`WithOperatorIDList` to name a Data Page that fetches a list of Remote Cases created by a specified user.

   c. In the **Apply to** field, select the class of the Remote Case Type in the consumer system.
   d. Click **Create and open**.
   e. On the **Definition** tab, in the **Data page definition** section, in the **Structure** field, select **List**.
   f. In the **Data Sources** section, in the **Source** field, select **Activity**, and then in the **Activity name** field, enter `pyFetchRemoteCaseInstancesWrapper`.
   g. On the **Parameters** tab, in the **Name** field, enter `DataTransformName`, and then in the **Description** field, enter the name of the Data Transform from step 3.
   h. Click **Save**.

©2025 Pegasystems Inc

> **Result:**
>
> You can use the Data Page in a query field or directly in a View. For more information, see Querying a Data Page.

Using the Data Page in a query field that you use in the Remote Case configuration:

7. Perform steps 1 through 10 in Querying a Data Page.

> **What to do next:**
>
> Configure a View to display a tab and a list of Remote Cases in the parent Case context. For more information, see Configuring a Full Page View.

Related concepts

- Accessing Remote Cases in the consumer application
- Using Data Pages in a Case Lifecycle

Related tasks

- Creating a Data Page

Error messages for Remote Case Types

If any issues connected with Remote Case Types occur, analyzing error messages helps you understand the cause of the issue and then make informed decisions about how to fix the issue. The JSON response to an issue includes relevant information that you can use when troubleshooting Remote Case Types.

JSON response

The following JSON response is common to all issues connected to Remote Case Types that might occur in your system:

©2025 Pegasystems Inc

```
{
  "errorClassification":"error classification",
  "localizedValue":"the root cause of the issue",
  "errorDetails":[
    {
      "message":"error message type",
      "erroneousInputOutputFieldInPage":"",
      "erroneousInputOutputIdentifier":"",
      "errorClassification":"",
      "localizedValue":"The error message that an application displays at run time.",
      "messageParameters":[

      ]
    }
  ]
}
```

The following JSON response includes the details of a sample execution error:

```
{
  "errorClassification":"Execution error",
  "localizedValue":"Exception while getting access token",
  "errorDetails":[
    {
      "message":"Error_Execution_Error",
      "erroneousInputOutputFieldInPage":"",
      "erroneousInputOutputIdentifier":"",
      "errorClassification":"",
      "localizedValue":"Error while executing request. Please contact administrator.",
      "messageParameters":[

      ]
    }
```

©2025 Pegasystems Inc

```
  ]
}
```

Logs in fixing issues

Ensure that you enable logs so that you can obtain additional details about issues that might occur in your system. For Remote Case Types, ensure that you enable `RemoteWebGatewayImpl` and `com.pega.platform.web.gateway.validations`. For more information, see Logging for VM-based deployments and containerized deployments on Kubernetes.

Incoming requests errors

The following table lists the errors that are related to processing incoming requests in the consumer system:

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
`Invalid tenant hash in the input URL`	Tenant hash in the URL of the producer system is invalid.	Check the tenant hash in the URL of the producer system that you provide when you create a Remote Case Type or update Remote Case Type details.
`Invalid request`	The AAT cookie or the JWT token is invalid.	Ensure that required credentials are present and correct in the request.
`Expired JWT Token`	The JWT token that you use is expired.	Enable logs and contact Pega Global Client Support.
`Empty remote system ID in the request`	The `remotesystemid` header is empty in the request.	Verify that the `remotesystemid` header is present in the request.

©2025 Pegasystems Inc

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
`Remote system details do not exist`	The `data-admin-remotesystem` data instance is empty in the `remotesystemid` element of the request.	Verify that the `data-admin-remotesystem` data instance that corresponds to the `remotesystemid` element exists in your system and is correct.
`AAT cookie not found in the request`	The AAT cookie is missing from the request.	Verify that the AAT cookie is part of the request.
`Invalid AAT`	The exception while processing the incoming AAT token occurred.	Enable logs, contact your security team, and contact Pega Global Client Support.
`Invalid URI request`	URI in the remote system details is invalid.	Verify that the URI of the remote system that you provide when you configure the consumer system is correct.
`Invalid request method`	The REST calls do not support the requested HTTP method.	Ensure that you invoke only the supported HTTP methods.
`Invalid CSRF Token`	The CSRF token is invalid.	Enable logs, contact your security team, and contact Pega Global Client Support.
`AAT Cookie is not parsed successfully`	The system is unable to parse the incoming AAT cookie.	Enable logs, contact your security team, and contact Pega Global Client Support.

©2025 Pegasystems Inc

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
SSL Handshake exception	TLS version in Authentication Profile is not compatible with the system.	Change **Lowest allowable SSL/TLS version** to **TLS version 1.2** in the remote system Authentication Profile.

Gateway processing errors

The following table lists errors that are related to gateway processing exceptions:

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
`Exception when write stream from remote response to servlet response`	Exception occurred while streaming a response from the remote system.	Enable logs and contact Pega Global Client Support.
`Exception while firing the remote request`	Not applicable.	Enable logs and contact Pega Global Client Support.
`Connection pool shutdown while firing the remote request`	The dynamic system setting for connection pool changed during request processing, and the request processing failed.	After you save the dynamic system setting with the new value, restart the request that failed.
`Exception while firing request to remote system`	Not applicable.	Enable logs and contact Pega Global Client Support.
`Exception while getting connection from connection pool`	The system failed to get the connection to fire the request.	Check logs for connection pools statistics and increase

©2025 Pegasystems Inc

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
		the connection pool size if necessary.
`Could not get token from remote system`	The token that the security settings return is empty.	Enable logs and contact your security team. Ensure that endpoints in the remote system run correctly.
`Exception while getting access token`	Security module returns an exception.	Enable logs, contact your security team and Pega Global Client Support.
`Exception while renewing access token`	Security module returns an exception on renewing access token.	Ensure that the client ID and client secret that you use to create Remote Case Types are correct. Ensure that the endpoint in the remote system is reachable. If the client ID, client secret, and the endpoint work correctly, contact Pega Global Client Support.
`Error while building socket factory for request system`	The `remotesystemid` element has an invalid configuration.	Ensure that the configurations for tenant, keystore, and truststore are correct. If the configurations are correct, contact Pega Global Client Support.

©2025 Pegasystems Inc

The root cause of the issue in JSON	Reason for the error message	Steps to fix the issue
`Authorization has failed at remote system`	The remote or producer system gave a 401 response.	Ensure that the client ID and client secret that you use to create Remote Case Types are correct. Ensure that the endpoint in the remote system is reachable. If the client ID, client secret, and the endpoint work correctly, contact Pega Global Client Support.
`Exception while reading post data from request`	An error with post data from the request occurred.	Verify the post that the request contains. If the data is correct, contact Pega Global Client Support.
`Exception while parsing request content type`	The request includes unsupported character encoding, or an exception occurred when parsing post data in the incoming request.	Verify the incoming request for unsupported character encoding. If the request does not include unsupported character encoding, contact Pega Global Client Support.

- Configuring dynamic system settings
- Troubleshooting tools and techniques
- Authentication
- Understanding authorized access tokens

PEGA ©2025 Pegasystems Inc

Creating Cases with a web embed

By embedding a Case into en external web page in the form of a web embed you can manage the business processes of users who use the external software. For example, you can embed a Case Type for job application processing on a web page that displays job offers.

> **NOTE:** Constellation replaces iframe-based mashups with div-based web embeds. For more information, see Center-out configuration in Constellation.

By creating web embeds you enable users to interact with Pega features and create Cases without having access to Pega Platform™ '24.2 software. For more information, see Embedding Constellation UI in web pages.

You can configure Case creation through a web embed by performing steps 8 through 16 in Creating web embed.

Related concepts

- Case Lifecycle elements
- Designing the UI in Case Types

Related tasks

- Configuring Case creation

Case Lifecycle elements

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

PEGA ©2025 Pegasystems Inc

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

- Stages that consist of Processes
- Processes that consist of Steps
- Steps that are either Automations or user actions

Stages

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

Stages in a Case Lifecycle

Processes

A Process consists of a series of tasks, or Steps, and visualizes a set of actions within a Stage. For example, for Application review Stage of hiring a candidate, you can add

©2025 Pegasystems Inc

Processes for Collecting personal information and Collecting documents, as shown in the following figure:

Sequential Processes in a Stage

By adding multiple Processes, you group tasks into logical phases and model the order in which Case workers complete tasks, because a Case moves to the next Process when all of the Steps from the previous Process are complete.

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

©2025 Pegasystems Inc

Parallel Processes in a Stage

Creating Processes saves time because you can reuse a Process in different Stages and Case Types. For example, you can reuse the Collecting documents Process from the reviewing job applications in other Case Types.

Steps

Steps are the smallest elements of a Case Lifecycle and represent single tasks or Assignments. A Step can be a user action or an Automation that an application performs. For example, for a Collect personal information Process, you can add a Send email Step, which is an Automation that sends a message to a job applicant, and a Collect information Step, which is an Action that a user performs, as shown in the following figure:

©2025 Pegasystems Inc

Steps in a Process

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **Stages in a Case Lifecycle**
- **Processes in a Case Lifecycle**
- **Steps in a Case Lifecycle**

Stages in a Case Lifecycle

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

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

- Submission
- Application review

©2025 Pegasystems Inc

- Interview
- Decision
- Job offer

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

> **NOTE:** As a best practice, use descriptive, business-oriented names for Stages, so that different stakeholders across your enterprise can understand the business value of each Stage. Typically, naming Stages is within the responsibility of business representatives.

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

The following figure presents a Case Lifecycle with Primary Stages and an Alternate Stage:

©2025 Pegasystems Inc

Case Designer displaying Primary Stages and an Alternate Stage

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

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

©2025 Pegasystems Inc

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

Naming convention for Stages

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

Populate your Case Lifecycle with Stages by completing the following tasks:

The following video shows how to add a Stage to your Case Type and what options you can configure for a Stage:

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **The Create Stage**

©2025 Pegasystems Inc

- **Creating a Primary Stage**
- **Creating an Alternate Stage**
- **Defining the entrance criteria for a Stage**
- **Configuring a Case resolution**
- **Defining Conditions for skipping a Stage**
- **Removing a Stage**
- **Best practices for Stages**
- Creating a Microjourney for customer success
- Setting Service-Level Agreements for Stages, Processes, and Steps

The Create Stage

To speed up Case creation and clearly visualize the elements that users interact with before Case processing starts, your Case Lifecycle includes a default Create Stage every time you create a new Case Type.

When you create a new Case Type, your Case Lifecycle includes the Create Stage by default. The Create Stage holds a View and Actions that users interact with to provide initial data before Case processing starts. For example, in an application for reviewing loan requests, you can define a View that displays fields to collect the customer's personal details and their requested loan amount.

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

©2025 Pegasystems Inc

Case Designer displaying a Case Type with a renamed Create Stage

In Constellation applications, the **Save and run** option opens the previewed Case without showing the Create form of the **Create** Stage in the dialog box first. Instead, the **Create** Stage is visible in the Case's action area when the Case opens.

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

- The Create Stage FAQ
- Adding the Create Stage to existing Case Types
- Technical considerations for the Create Stage in existing Case Types

Related concepts

- Case Types

Related tasks

- Processes in a Case Lifecycle

©2025 Pegasystems Inc

Related reference

- Best practices for Stages

The Create Stage FAQ

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

What does the Create Stage include?

The Create Stage includes a default Create Process with a multi-step form. You can modify the Stage to meet your business needs by adding Processes and Steps. For greater flexibility, you can also rename the Create Stage and the Create Process to unique names that fit your business scenario.

The following figure shows a Create Stage that is renamed to `Submission`, which includes the default Create Process renamed to `Collect information`, with a Multi-step Form. The Create Stage in the example also includes an additional `Collect documents` Process with two automated Steps:

©2025 Pegasystems Inc

The Create Stage

For more information about Stages and Processes, see Case Lifecycle elements.

When does the processing of Cases with the Create Stage start?

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

What happens if a user cancels or abandons a Case in the Create Stage?

The way that a Case behaves when it is canceled in the Create Stage depends on how the system initiates the Case:

©2025 Pegasystems Inc

Scenario	Behavior
Case created using the + button	The Case is deleted from the database, including its history entry. The Case ID iterator is incremented.
Case created through a Local Action in another Case	The Case is deleted from the database, including its history entry. The Case ID iterator is incremented.
Case created through an Automation in another Case	The Case is not deleted. The Assignment remains available. This behavior may depend on configuration.

If a Case is abandoned, for example when the user closes the browser tab or the session times out, it can remain in `New` status, depending on the system configuration. To ensure auditability and avoid clutter from abandoned Cases, consider implementing a custom cleanup policy or tracking mechanism. For more information, see Case archiving policy.

How does the Create Stage work with creating a Case by email?

When creating a Case by email, at run time, your application omits the default Create Process. The purpose of the Create Process is to capture the initial data that is required before submitting a Case, but because an email starting a Case meets the same requirement, the application skips the Create Process. For more information, see Enabling creation of Cases by email.

Can I use the Create Stage for different Channels?

To reach wider audiences, you can create a channel-specific Process inside the Create Stage, for example, a process for users who interact with your application through a chat bot. For more information, see Adding a conversational Channel to a Case Type Process.

PEGA ©2025 Pegasystems Inc

Do all my Case Types include the Create Stage by default?

All Case Types that you create in Pega Platform version 8.5 and later have the Create Stage by default. Case Types that you migrate from Pega Platform version 8.4 and earlier do not include the Create Stage. You can add the Create Stage to your migrated Case Types to take advantage of all the possibilities that the Create Stage offers. For more information, see Adding the Create Stage to existing Case Types.

Can I remove the Create Stage from my Case Type?

The Create Stage is a mandatory element for collecting information before a user submits a Case, so you cannot remove the Create Stage from a Case Lifecycle.

Can I configure entry Validation for the Create Stage?

A Case always enters the Create Stage, so you do not need to configure entry Validation. If you want to configure Validations upon Case creation, configure Validations as part of the View for the first Assignment in the Create Stage. For more information, see Validating input values on a form.

If the Create Stage in your business scenario includes no Assignments, configure Validation for the first Stage after the Create Stage. For more information, see Defining the entrance criteria for a Stage.

Can I route an Assignment in a Create Stage to a Work Queue?

The main purpose of the Create Stage is to collect initial data before Case processing, so configure routing of the Assignments in the Create Stage to a current user. If your business scenario requires routing Assignments to a Work Queue or a user different that a current user, configure routing logic on a Stage that follows the Create Stage. For more information, see Creating a Work Queue and Defining work routing settings for an operator.

PEGA ©2025 Pegasystems Inc

Can I configure parallel Processes in the Create Stage?

Yes, you can configure parallel Processes in the Create Stage. As a best practice, configure your application to route parallel Processes to a current user, because the purpose of the Create Stage is to collect data before Case processing begins. For more information, see Adding a parallel Process to a Stage.

Can a user navigate back to the Create Stage?

No, a Case can only enter the Create Stage once.

How do parent and child Case Types interact when the child includes a Create Stage?

When a child Case is created from a parent Case, the **Cancel** button in the child Case behaves differently than in standalone Cases. Instead of deleting the Case, **Cancel** closes the Assignment and returns control to the parent Case. This behavior also applies to sibling Cases created through Local Actions or Automations. To customize this interaction, define conditional logic in the Process or use post-processing Actions to manage transitions. For more information, see Conditionally starting a Process.

Can I start a Case on the Stage that follows the Create Stage?

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

- Enabling field-level auditing
- Configuring Case creation
- Completing work on time
- Attaching content to a Case

PEGA ©2025 Pegasystems Inc

Adding the Create Stage to existing Case Types

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

When you update your system from Pega Platform 8.4 or earlier, you can include the Create Stage in your existing Case Types.

By adding the Create Stage to your Case Lifecycle, you clearly visualize the Views and actions that users interact with before submitting a Case. If you choose not to add the Create Stage, your application is still fully backward compatible.

After the update, your Case Types work correctly and rely on the *pyStartCase* starting Process. Your Case Types become independent of the starting Process only after you add the Create Stage. Adding the Create Stage is optional and requires additional configuration, however, the Create Stage helps you define your Case Types and develop your application faster.

To fully understand the impact of adding the Create Stage to your existing Case Types, analyze the technical considerations. For more information, see Technical considerations for the Create Stage in existing Case Types.

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 header of the **Case Type** work area, click **Actions** > **Add Create Stage**.
3. In the **Add Create Stage** window, analyze the changes that occur after you add the Create Stage, and then click **Submit**.
4. Click **Save**.

©2025 Pegasystems Inc

Result:

If you configured your Case Type to skip the Create View, the system marks the first Stage in your Case Lifecycle as the Create Stage, and displays the View associated with the Stage as the Create View. If you configured your Case to display the Create view, the system adds a new Create Stage as the first Stage, and you need to define the Create view. For more information about the characteristics of Case Types with the Create Stage, see The Create Stage.

What to do next:

If your Case Type has no Create View configured, define fields that users need to interact with before submitting the Case for processing. For more information, see Creating Views for Case Types.

Related concepts

- Case Types

Related tasks

- Processes in a Case Lifecycle

Technical considerations for the Create Stage in existing Case Types

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

The Create Stage in existing Case Types

To understand how the Create Stage can impact your existing Case Types, consider the following scenarios:

©2025 Pegasystems Inc

- If you configured your existing Case Type to skip the Create View when users create a Case, Case Designer marks the first Stage in your Case Type as the Create Stage by adding a green bar to the Stage header.
- If you configured your existing Case Types to display the Create View, Case Designer adds the Create Stage as the first Stage in your Case Type.
- Your Case Types are independent of the *pyStartCase* or any other starting Process. As a result, any security settings or other configurations that you define on the starting Process Rule are inapplicable to the Case Type.
- If you configured your Case Type to create temporary Cases, the option is not applicable after adding the Create Stage. All the Case Types that include the Create Stage create persistent Cases.
- When you configure the Create Stage in a Case Type, the system immediately creates a Case every time you enter the Create Stage. If you do not want to include Cases that are in the Create Stage or have the `Resolved-Cancelled` status in your dashboards or reports, update the relevant report definitions to filter out such Cases.

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

The Case Type Rule has only *pyStartCase* configured as a starting Process without modifications

Consider the following circumstances:

- You need to configure the View for the Create Step in the newly added Create Stage. For more information, see Designing the UI in Case Types
- The only Data Transform that works upon creation of a Case with the Create Stage is *pyDefault*. If you configured another Data Transform to work upon a Case creation in your existing Case Type, Pega Platform automatically changes the Data Transform to *pyDefault*. To apply your custom configurations, call your Data Transform from the *pyDefault* Data Transform or configure a Process in the Create Stage to transform data.

©2025 Pegasystems Inc

- If you configured any Validations on Case creation, you need to reconfigure the Validations to apply to the next Stage after the Create Stage.

The Case Type Rule has only *pyStartCase* configured as a starting Process with modifications or the Case Type Rule has many starting Processes configured

Consider the following circumstances:

- You need to configure the View for the Create Step in the newly added Create Stage. For more information, see Designing the UI in Case Types
- You need to move any pre-processing or post-processing logic from the *pyStartCase* starting Process to the Create Stage.
- The only Data Transform that works upon creation of a Case with the Create Stage is *pyDefault*. If you configured another Data Transform to work upon Case creation in your existing Case Type, Pega Platform automatically changes the Data Transform to *pyDefault*. To apply your custom configurations, call your Data Transform from the *pyDefault* Data Transform or configure a Process in the Create Stage to transform data. For more information, see Data Transforms.
- If you configured any Validations on Case creation, you need to reconfigure the Validations to apply to the next Stage after the Create Stage.

Related concepts

- Case Types

Related tasks

- Processes in a Case Lifecycle

©2025 Pegasystems Inc

Creating a Primary Stage

Represent main phases of your business process by creating Primary Stages in a Case Lifecycle. 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.

For example, for a Review job application Case Type, you can create Stages, such as Submission, Application review, Decision, and Job offer.

Before you begin:

Define a Case Type. For more information, see Case Types.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

When you create a new Case Type, the system automatically creates the first Stage, which is the **Create** Stage. For more information, see The Create Stage.

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 **Lifecycle**.
3. In the **Case Lifecycle** section, click **Stage**, and then enter a unique name in the text box.

> **NOTE:** As a best practice, use descriptive, business-oriented names for Stages, so that different stakeholders across your enterprise can

©2025 Pegasystems Inc

understand the business value of each Stage. Typically, naming Stages is within the responsibility of business representatives.

4. In the **Stage** property pane, on the **General** tab, define the transition after this Stage is complete:
   - To move to the next Stage, select **Automatically move to next Stage**.
   - To let the Case worker 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.
5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

The following figure shows a Case Type with four empty Stages. A green bar marks the **Create** Stage, and a red bar marks the **Resolution** Stage:

©2025 Pegasystems Inc

Stages in a Case Type

At run time, a breadcrumb navigation of a Case displays names that you associate with the Stages, as shown in the following figure:

Case breadcrumb navigation

What to do next:

Organize work within Stages by adding Processes. For more information, see Processes in a Case Lifecycle.

- Case Lifecycle elements
- Creating an Alternate Stage
- Processes in a Case Lifecycle

Creating an Alternate Stage

Ensure that you can resolve a business process when an exception occurs by adding an Alternate Stage to the Case Lifecycle. For example, you can use an Alternate Stage to process 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 **Lifecycle**.
3. Create an Alternate Stage:
   - To create a new Alternate Stage, in the **Case Lifecycle** section, click **Alternate Stage**.
   - To convert a Primary Stage to an Alternate Stage, on the menu next to the Stage name, click **Set as Alternate Stage**.
4. Decide how the Case proceeds after all Processes in this Stage are complete:

©2025 Pegasystems Inc

- To let the Case worker decide which Stage the Case enters after this Stage, in the Stage property pane, select **Wait for a user action**.
- To resolve the Case, in the Stage property pane, select **Resolve the Case**, and then configure details of Case resolution. For more information, see Configuring a Case resolution.

5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a Case Lifecycle of reviewing a job candidate with four Primary Stages and an Alternate Stage. A Case enters the Alternate Stage when a hiring manager rejects the candidate in the Review Process.

An Alternate Stage

©2025 Pegasystems Inc

What to do next:

Populate the Stage with Processes. For more information, see Processes in a Case Lifecycle.

- Case Lifecycle elements
- Creating a Primary Stage
- Processes in a Case Lifecycle

Defining the entrance criteria for a Stage

Prevent delays and improve the efficiency of the Case resolution process by verifying that the user attaches relevant documentation and provides appropriate information. Validate Case properties, attachments, and conditions to determine whether a Case can enter a specific Stage as it moves through the Case Lifecycle.

For example, you can check whether a vehicle loan Case has an attached credit score before sending the Case to a finance manager for approval.

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

- Stages in a Case Lifecycle
- Case Lifecycle elements
- Creating a Primary Stage
- Attachment types
- Controlling access to Case attachments
- Categorizing Case attachments

Setting required attachments for Stage entry

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

©2025 Pegasystems Inc

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

Before you begin:

Add attachment categories that reflect your business needs to your Case Type and ensure that you select the **Required for resolution** checkbox in step 5 in Categorizing Case attachments.

NOTE: You cannot define entrance criteria for the Create Stage. For more information, see The Create Stage.

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

   For example:

   The following figure shows a design-time configuration of a Stage with a required attachment. Users that process a Case have to provide an attachment of the selected category to submit the Case.

©2025 Pegasystems Inc

Setting a required attachment for Stage entry

6. Click **Save**.
7. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

Result:

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

Validating input values for Stage entry

Ensure that a Case contains relevant data before it enters a specific Stage by validating the input values that users provide. By validating the input values for a Stage entry, you prevent the Case from proceeding to the next Stage without the relevant information. If the value that the user enters into a form at run time meets the Condition in a Validate Rule, the application displays an error message.

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

Before you begin:

Create a Case Type. For more information, see Case Types.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

©2025 Pegasystems Inc

> **NOTE:** You cannot define entrance criteria for the **Create** Stage. For more information, see The Create 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 **Data Model** tab, click **Validations**.
3. Select the cell that corresponds to the row of the field and the column of the Stage for which you want to perform Validation:
   - To define Validation for a single-value field, click **Add entry Validation**.
   - To define Validation for a data relationship field, click **Click to configure**, and then click **Add entry Validation**.
4. In the **Stage entry Validation** dialog box, define the Validation Conditions:

Choices	Actions
Define a Validation Condition for a field	a. In the list of values, select **Fields**, and then select the name of the field that you want to validate. By default, the system provides the field that you select in step 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 to the user input.
Apply an existing When Condition as a Validation Condition	a. In the list of values, select **When Conditions**, and then select the name of the When Condition that you want to apply.

©2025 Pegasystems Inc

Choices	Actions
	b. In the comparator list, select the test that you want to perform on the When 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, the application displays an error message.

5. **Optional:** To create a Condition with multiple Validation parameters, add a logical operator to the Condition:

Choices	Actions
The Condition passes when all properties meet the criteria	a. Click Add a row. b. In the operator list, select and. c. In the list of fields, select the name of the field that you want to validate or the name of the When Condition that you want to apply. d. In the comparator list, select the test that you want to perform on the field or on the When Condition. e. In the value field, enter or select a value to compare to the user input. The `Select values` option lists the fields and values that you can use in the Condition.

©2025 Pegasystems Inc

Choices	Actions
The Condition passes when any of the properties meet the criteria	a. Click Add a row. 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 or on the When Condition. e. In the value field, enter or select a value to compare to the user input. The Select values option lists the fields and values that you can use in the Condition.

For example:

In a recruitment form, the user indicates 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.

6. In the **Stage entry Validation** window, in the Then display error message as field, enter the text to display when user input meets the Validation Conditions at run time.
7. **Optional:** To select additional Stages for which you want to perform the Validation, in the To validate Stage entry for the Stages parameter, select the Stages.
   By default, the system selects the checkbox for the Stage that you select in step 3.
8. Click Submit.

©2025 Pegasystems Inc

9. **Optional:** To define more Validation Conditions, in the cell in which you want to add more Conditions, click **Add another Validation**, and then repeat steps 4 through 8.
10. Click **Save**.
11. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

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

For example:

Configuring a Case resolution

Configure a resolution method in a Stage to control the final status and processing that occurs when a Case reaches the end of the Case Lifecycle.

For example, you can configure a Loan request Case Type to have two possible resolution statuses: `Resolved-MoneyDisbursed` and `Resolved-InvalidClaim`. To reach the `Resolved-MoneyDisbursed` status, you configure the Case to follow the Primary Stage sequence. To reach the `Resolved-InvalidClaim` status, you redirect the Case to an Alternate Stage.

PEGA ©2025 Pegasystems Inc

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 Stage at which you want to resolve the Case.
   Typically, the last Primary Stage in the Case Lifecycle or an Alternate Stage resolves a Case.
3. In the **Stage** property pane, on the **General** tab, click **Resolve the Case**.
4. **Optional:** To change the default final status for the resolved Case, in the **Resolution status** field, define the new status:
   - To use a different existing status, select a status value from the list.
   - To create a new status, enter a unique name by using the format `Resolved-[<your status name>]`.

   > **For example:**
   >
   > Enter `Resolved-MoneyDisbursed`.

   By default, the resolution status that the Case displays when the Stage processing ends is `Resolved-Completed`. For more information, see Case Status values.
5. **Optional:** To delete any open Assignments from the previous Stages, select the **Delete open Assignments** checkbox.
6. **Optional:** To clean up child Case dependencies when the current Case reaches its resolution, select the **Resolve open child Cases** checkbox.
7. **Optional:** To change the default final status for the resolved child Cases, in the **Resolution status of child Cases** field, select the new status:
   - To use a different existing status, select a status value.
   - To create a new status, enter a unique name by using the format `Resolved-[<your status name>]`.

   > **For example:**
   >
   > Enter `Resolved-Withdrawn`.

   By default, the resolution status of the related child Cases is `Resolved-Unspecified`.

©2025 Pegasystems Inc

8. Click **Save**.
9. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

When all the Steps in the Stage reach completion, the current Case changes status to the value configured in the **Resolution status** field.

If you configure child Case resolution in steps 6 and 7, the related child Cases also reach resolution and their status changes to the value configured in the **Resolution status of child Cases** field.

For example:

The following figure shows details of a resolved Case with a newly created status, `Resolved-MoneyDisbursed`:

©2025 Pegasystems Inc

Case Status upon resolution

- Defining a Case Status
- Case Status values
- Best practices for defining Case Statuses
- Stages in a Case Lifecycle
- Case Lifecycle elements
- Identifying Case Type elements
- Building Case Types

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.

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 **Lifecycle**.
3. In the **Case Lifecycle** section, click a Stage name.

©2025 Pegasystems Inc

4. In the **Stage** property pane, on the **General** tab, define a 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:** `Work 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. **For example:** The following figure shows two Conditions with the `or` operator. Ar tun time, the application skips the Stage if any of the Conditions evaluate to true.

©2025 Pegasystems Inc

Choice	Steps
	 Conditions for skipping a Stage 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 Create Stage and last Stage in the sequence.

5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, the system evaluates the Condition and skips the Stage if the value meets the Condition.

©2025 Pegasystems Inc

- Case Lifecycle elements
- Stages in a Case Lifecycle
- The Create Stage
- The Skipping a Process or a Stage module on Pega Academy

Removing a Stage

Deliver a flexible application that you can conveniently adjust to dynamically changing circumstances by removing Stages that are no longer relevant to your business requirements. As a result, you save time because you can edit an existing Case Type instead of creating a new one from scratch.

For example, if the process for hiring new employees in your organization changes and one of the Stages in your hiring Case Type is no longer required, you can simply remove the Stage to reflect the change and adjust the existing Case Type to the new hiring process.

You can delete Primary Stages that visualize the happy path of your business process, as well as Alternate and Resolution Stages that correspond to actions that occur if your business process diverts from the primary path. Removing a Stage does not delete any resources that you connect to the Stage, such as Service-Level Agreements or Notifications. The resources are present in your system but do not participate in Case processing anymore. You can still reuse these resources in other elements of your application.

NOTE:

- You cannot delete the Create Stage.
- Ensure that the new workflow design does not cause workflow issues for existing Cases at run time. For more information, see Handling flow changes for Cases in flight.

©2025 Pegasystems Inc

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 Stage header, click **More**, and then click **Delete Stage**.
3. Click **Save**.
4. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

The Stage is removed from the Case Lifecycle. The following image shows an example of removing a Stage:

Removing a Stage

- The Create Stage

©2025 Pegasystems Inc

- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools
- Case Lifecycle elements
- Finding Rules by type

Best practices for Stages

Stages define the major phases of work in a Case Type and provide a clear structure for Case progression. Follow these best practices for configuring Case Stages to ensure that your application is easy to understand, maintain, and use.

Primary Stages

Primary Stages represent the standard path that the majority of Cases follow toward a successful outcome. These Stages define key milestones in typical Case processing.

You can identify the Primary Stages of a Case by answering the following questions:

- Should the majority of Cases of this type go through this Stage?
- Would completing the Steps in this Stage be considered part of the usual Case processing?
- Do the Processes in this Stage represent a key component of the Case Lifecycle?
- Is going through this Stage required to reach a positive Case resolution?

Best practices for Primary Stages:

- Use noun-based or noun phrase names to represent a state or milestone in the Case, such as **Request**, **Review**, or **Confirmation**.
- Avoid using verbs or action phrases like **Requesting** or **Confirm travel**, which can be harder to read naturally in system messages and reports.
- Choose names that work naturally in the following system-generated sentences:
  - This Case is in <*Stage name*>.
  - How many Cases are in <*Stage name*>?
  - When will this Case move to <*Stage name*>?

©2025 Pegasystems Inc

- Define between four and seven Primary Stages to balance clarity and completeness.
- Ensure that Primary Stages follow a logical progression from start to resolution.

For example, an application has a Case Type for employee travel requests, which includes four Primary Stages:

1. **Request**, which collects travel destination details.
2. **Review**, which sends the request for manager approval.
3. **Booking**, which books the travel.
4. **Confirmation**, which represents the outcome, indicates that the reservation is complete, and sends a confirmation notification.

The transition from the **Request** to the **Review** Stage occurs when sufficient travel data is collected from the employee to send the request to a manager for review. The transition from **Review** to **Booking** is based on approval - a human decision by a manager - to allow the Case to continue. The transition from **Booking** to **Confirmed** occurs automatically after the travel agent has completed the reservations.

For more information, see Creating a Primary Stage.

Alternate Stages

Alternate Stages supplement Primary Stages by handling exceptions, alternate paths, and negative resolutions. Alternate Stages are optional and used only when applicable.

You can identify the Alternate Stages of a Case by answering the following questions:

- Does this Stage represent a negative resolution?
- Is this an exception or error scenario that only a minority of Cases will go through?

Best practices for Alternate Stages:

- Use Alternate Stages for manual interventions or negative outcomes, for example, for exceptions or rejections.

PEGA ©2025 Pegasystems Inc

- Keep names concise and descriptive, for example **Withdrawn** or **Booking Exception**.
- If the exception can be resolved, allow the Case to re-enter the Primary path at the appropriate point.

For example, a travel request Case includes the following Alternate Stages:

- **Withdrawn** when the originator cancels the request.
- **Rejected** when the manager denies approval.
- **Booking Exception** triggered when automated booking fails and requires manual handling.

For more information, see Creating an Alternate Stage.

Resolution Stages

Resolution Stages indicate that the Case will finish its lifecycle upon conclusion of the Stage. These Stages represent Case conclusions, both successful and unsuccessful.

Each Case Type should have at least one Resolution Stage, usually the final Primary Stage. Additional Resolution Stages can also exist in Alternate Stages.

Best practices for Resolution Stages:

- Use noun-based names that describe final Case outcomes, such as **Completion**, **Rejection**, **Withdrawal**, **Fulfillment**, or **Resolution**.
- Configure a corresponding resolution status for each Resolution Stage, such as **Resolved-Completed**, **Resolved-Withdrawn**, and **Resolved-Rejected**.

For more information, see Configuring a Case resolution.

Stage transitions

Moving from one Stage to another signifies progression through the Case Lifecycle. Transitions can be automatic or based on user actions.

PEGA ©2025 Pegasystems Inc

Best practices for Stage transitions:

- Base Stage transitions on clear conditions:
  - Automatic - when the Case automatically captures all necessary data or a system event triggers the transition.
  - Manual - when the Case assigns a user with a clear action to make a decision or approve the Case.
- When configuring a Stage, choose the appropriate transition option:
  - **Automatically move to next Stage** if all required data is collected or a business event advances the Case, for example, waiting for a child Case.
  - **Wait for a user action** if one or more Case-wide Actions or optional Processes must be completed in addition to the Stage Steps before the Case can move to the next Stage.
  - **Resolve the Case** if the Case reaches a final outcome.

For more information, see Automatically moving a Case to a different Stage.

Related concepts

- The Create Stage

Processes in a Case Lifecycle

Processes in Stages organize related tasks in your business Case. With Processes, you can control the order of events in your Case, in addition to who performs the work and in what manner.

For example, for the Application review Stage of a recruitment Case, you can add the Conduct interview and Review collected information Processes.

When you add Processes to a Case Type, you organize related tasks in a logical way, instead of having a list of loosely connected tasks. You also define an order of Case events, so that a Case can only move to the next Process after completing the Steps in the current Process. For example, when reviewing a job candidate, a Process that

PEGA ©2025 Pegasystems Inc

includes a job interview can only start after a Process for collecting all required documents is complete.

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

Sequential Processes in a Stage

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

NOTE: As a best practice, use descriptive, business-oriented names for Processes, so that different stakeholders across your enterprise can

©2025 Pegasystems Inc

> understand the business value of each Process. Typically, naming Processes is within the responsibility of business representatives.

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

A Stage that includes two parallel Processes

Populate your Case Lifecycle with Processes by completing the following tasks:

- **Adding a sequential Process to a Stage**
- **Adding Steps to Processes**
- **Adding a parallel Process to a Stage**
- **Defining Conditions for starting a Process**

©2025 Pegasystems Inc

- **Conditionally starting a Process**
- **Configuring navigation to a previous Step**
- **Removing a Process from a Stage**
- **Draft mode of Case Processes**

- Creating a Microjourney for customer success
- Removing a Process from a Stage
- Configuring complex Processes

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 and Creating an Alternate Stage.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

©2025 Pegasystems Inc

For more information, see Creating a new application from a Blueprint.

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 Lifecycle** section, hover over a Stage, and then select a Process to add:
   - To add a new Process, click **More** > **Add Process** > **New Process**.
   - To reuse an existing Process, click **More** > **Add Process** > *Process name*.
4. In a new text field, replace the default Process label with a descriptive name.

NOTE: As a best practice, use descriptive, business-oriented names for Processes, so that different stakeholders across your enterprise can understand the business value of each Process. Typically, naming Processes is within the responsibility of business representatives.

5. **Optional:** To change the run-time order of a Process, drag the Process to a different position in the Stage.
6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

PEGA ©2025 Pegasystems Inc

Result:

At run time, the Process starts when all Steps in the previous Process are complete.

What to do next:

- Provide meaningful tasks for your Process by adding Steps. For more information, see Adding Steps to Processes.
- Make your Process applicable only to a certain scenarios by defining Conditions to start a Process. For more information, see Conditionally starting a Process.

- Conditionally starting a Process

Adding Steps to Processes

Model your business process with basic tasks that users or automations resolve by adding Steps to your Processes. 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.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

©2025 Pegasystems Inc

For more information, see Creating a new application from a Blueprint.

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.
   - To add a Step that displays a form that users complete to provide data, click **Collect information**, and then perform steps 4 through 10 in Collecting information from a user.
   - To add an approval Step, click **Approve/Reject**, and then perform steps 4 through 9 in Requesting approval from users .

     NOTE: Adding an Approval Step automatically creates an Alternate Stage in a Case Lifecycle. For more information, see Stages in a Case Lifecycle and Creating an Alternate Stage.

   - To add a Step that creates multiple screens with which users interact to provide data, click **Multi-step Form**, and then perform steps 4 through 10 in Adding a Multi-step Form to a Stage.
   - To add an existing Process as a single Step, click **More** > **Processes**, and then select the Process that you want to use.

     For more information, see Calling one Process from another Process.

   - To add a Step that requires human interaction, click **More** > **User action**, and then select a user action that you want to use.

     For more information, see Configuring and working with optional Actions in Cases.

©2025 Pegasystems Inc

- To add an Automation that an application performs, click **More** > **Automations**, and then select an Automation that you want to use.

  For more information, see Adding an automated Step to a Process.

4. Click **Save**.
5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a Stage in a Case Type of reviewing a job candidate. The Stage includes one Process that has three Steps of different kinds: an Automation that pauses the Process until a user conducts a job interview, a Step to collect information, and an Approval Step.

©2025 Pegasystems Inc

Sample Steps

What to do next:

Start planning your Case Type by associating Personas with your Case Type. For more information, see Adding Personas to organize users.

Adding a parallel Process to a Stage

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

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

©2025 Pegasystems Inc

Another common use case for parallel processing is managing multi-channel applications. In this scenario, a parallel process is conditionalized to be channel-specific. For example, the candidate in a recruitment process can send out their application through a process that can only run in a self-service channel.

For relevant training materials, see the Parallel Processing module on Pega Academy.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Workflow** tab, click **Lifecycle**.
3. In the **Case Lifecycle** 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 Multi-step form, click **More** > **Add parallel Process** > **New Multi-step 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.

> **NOTE:** As a best practice, use descriptive, business-oriented names for Processes, so that different stakeholders across your enterprise can understand the business value of each Process. Typically, naming Processes is within the responsibility of business representatives.

5. **Optional:** To manage multi-channel Processes, in the **Process** pane, define the conditions for limiting the Process to a given other Channel.

> **For example:**
>
> Limit the Process to mobile devices. For more information, see Conditionally starting a Process. The following image shows a sample configuration:

©2025 Pegasystems Inc

Conditional Process start for mobile devices

6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, the Process starts when a Case enters the Stage.

For example:

The following figure shows a Stage that includes two parallel Processes. A hiring manager can conduct a job interview while an HR worker reviews documents.

©2025 Pegasystems Inc

Parallel Processes

What to do next:

Add meaningful tasks to your parallel Process by adding Steps. For more information, see Steps in a Case Lifecycle.

- Conditionally starting a Process

Defining Conditions for starting a Process

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

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

©2025 Pegasystems Inc

For relevant training materials, see the Skipping a Process or a Stage module on Pega Academy.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, click a Process.
3. In the **Process** property pane, on the **General** tab, define the Conditions to start the Process:

Choices	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:** `Work status is equal to "New"` d. **Optional:** To add more Conditions, click **Add a row**, and then repeat step 3.c. e. **Optional:** To group the Conditions, select comparators from the list.

©2025 Pegasystems Inc

Choices	Actions
	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**.
5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, the system evaluates the Conditions and starts the Process when the Conditions return true values.

Related tasks

- Changing the path of a Process

©2025 Pegasystems Inc

Conditionally starting a Process

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

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

You can define conditions to start a Process both in App Studio and Dev Studio. In App Studio, you can reuse existing conditions or create new ones by using a Condition Builder, while in Dev Studio you can use a When Rule or an Expression to define when the Process starts.

For relevant training materials, see the Skipping a Process or a Stage module on Pega Academy.

1. Open the Process that you want to configure:
    - To configure the Process in App Studio, in the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
    - To configure the Process in Dev Studio, in the navigation pane of Dev Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, click the Process that you want to configure.
3. In the **Process** pane, on the **General** tab, define Conditions to start the Process:

©2025 Pegasystems Inc

Choices	Actions
Always start a Process	In the Start Process (otherwise skip) list, select Always.
Define a new Condition in App Studio	a. In the Start Process (otherwise skip) list, select Custom Condition, and then click the Configure Conditions icon. b. In the **Configure Condition** window, define a condition by selecting a property, comparator, and value to evaluate against the property at run time. **For example:** To start a Process only when a job candidate applies for a manager role, select `Position type equals to "Manager"`. As a property, you can select a field or a When Condition. For more information about building Conditions, see Defining Conditions in the Condition builder. c. **Optional:** To define more conditions, click Add a row, and then repeat step 3.b. d. If you have multiple conditions, specify whether all or any conditions need to evaluate to true at run time

©2025 Pegasystems Inc

Choices	Actions
	by selecting either **and** or **or** between the conditions. **For example:** The following figure shows the Conditions to set if you want a Process to start when a job candidate applies for a manager role or when the work experience of the candidate is less than three years: *Conditions for starting a Process* e. Click **Submit**.
Use an existing Condition in App Studio	a. In the **Start Process (otherwise skip)** list, select **Existing Condition**. b. In the list of Conditions, select a Condition that you want to evaluate, and then decide whether the Condition needs to evaluate to true or false to start the Process. **For example:** The following figure shows the Condition that starts a

©2025 Pegasystems Inc

Choices	Actions
	Process if a user starts your application on a mobile device: *Reusing an existing Condition*
Use a When Rule in Dev Studio	a. In the Start Process list, select When Rule. b. In the text field, press the Down arrow key, and then select the When Rule that you want to use. **TIP:** You can also create a new When Rule by clicking the Open icon. For more information, see Creating a When Rule.
Use an Expression in Dev Studio	a. In the Start Process list, select Expression. b. In the text field, press the Down arrow key, and then select the Expression that you want to use.

©2025 Pegasystems Inc

Choices	Actions
	TIP: You can also create a new Expression by clicking the **Build an Expression** icon.

4. Click **Save**.
5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **Result:**
>
> At run time, when a Case reaches the Process, an application evaluates the Conditions and, based on the results, either starts or skips the Process.

- Processes in a Case Lifecycle
- Displaying supporting Processes Conditionally
- Displaying optional Actions conditionally

Configuring navigation to a previous Step

Configure a navigation button to go back to a previous Step in a Process. The possibility to see and modify previous Steps allows for greater flexibility when performing a task.

You can configure the back navigation button only within a single Process, and not between Processes. However, the navigation is enabled within all Subprocesses as well.

©2025 Pegasystems Inc

You can also configure the button to go to the first or last Step of a Subprocess. For more information, see Adding additional configuration to a Subprocess.

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 Lifecycle** section, click the Process that you want to configure.
3. In the **Process** pane, on the **General** tab, select the **Allow users to go back to the previous Step** checkbox.
4. Click **Save**.
5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, the user can use the **Previous** navigation button to go back to a previous Step in a Process and all Subprocesses, as seen in the following figure:

©2025 Pegasystems Inc

Previous Step navigation

- Configuring Stages on a Case Type Rule form
- Adding additional configuration to a Subprocess
- Creating a Case Type Rule form in Dev Studio

Removing a Process from a Stage

To save time and resources, adjust your existing Case Types to changing business scenarios by removing Processes that are no longer required. Modifying Case Types helps you efficiently use resources in your application and accurately respond to dynamic situations.

In a sample scenario, if your organization no longer requires job candidates to deliver printed proofs of education, you can remove a Process that creates the Steps necessary to collect such documents.

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

NOTE: Ensure that the new workflow design does not cause workflow issues for existing Cases at run time. For more information, see Handling flow changes for Cases in flight.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. Hover over a Process name in a Stage.
3. Click the **Delete** icon.
4. Click **Save**.

©2025 Pegasystems Inc

5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Calling one Process from another Process
- Creating a stand-alone Process

Draft mode of Case Processes

When you add a Process to your Case Lifecycle, by default the Process is in draft mode. Draft mode provides an option to test the Case Type before you advance with your application development, even if Processes in your Case Lifecycle contain configuration issues.

For example, you can test how a Loan request Case Type behaves at run time even before you configure the Views with fields that users use to provide information.

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

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

PEGA ©2025 Pegasystems Inc

Warning about errors in Processes

Window with a list of errors

Related concepts

- Configuring complex Processes

Related tasks

- Configuring Case Type settings
- Managing Case Types

©2025 Pegasystems Inc

Steps in a Case Lifecycle

A Step in your Case Lifecycle represents a single Assignment in your business process. By adding a Step, you define a task that a user, your application, or an external application performs to move a Case closer to resolution.

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

> **NOTE:** As a best practice, use descriptive, business-oriented names for Steps, so that different stakeholders across your enterprise can understand the business value of each Step. Typically, naming Steps is within the responsibility of business representatives.

The following figure presents a set of related Steps:

©2025 Pegasystems Inc

Visualization of actions in a business process

When you visualize your work by adding Steps to a Case Lifecycle, you clearly communicate what actions your business process requires to reach a resolution, and you organize your work in the most efficient way.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **Step types**
- **Collecting information from a user**

©2025 Pegasystems Inc

- **Assigning tasks to users**
- **Adding instructions to Assignments**
- **Adding a Multi-step Form to a Stage**
- **Requesting approval from users**
- **Adding an automated Step to a Process**
- Creating a Microjourney for customer success
- Processes in a Case Lifecycle

Step types

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

In a sample scenario, you can use Steps in a **Mortgage** Case Type to retrieve a customer's financial history, prompt a Case worker to calculate a new interest rate, and notify a Case manager when the terms of the loan are ready for approval. You choose the Step types when you define the Case Lifecycle.

> **NOTE:** As a best practice, use descriptive, business-oriented names for Steps, so that different stakeholders across your enterprise can understand the business value of each Step. Typically, naming Steps is within the responsibility of business representatives.

Pega Platform™ supports the following Step types:

©2025 Pegasystems Inc

Collect information

Assigns a task, which you define in a form with fields, to a user or Work Queue. For example, in a **Loan** request Case Type, you can add a Step to collect users' personal details.

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

Collect information Step

For more information, see Collecting information from a user.

©2025 Pegasystems Inc

Approve/Reject

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

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

Approve/Reject Step

For more information, see Requesting approval from users.

Processes

Calls another Process from the current Process so that you can reuse an existing set of Steps instead of repeatedly adding and configuring multiple Steps. For example, you can reuse a Process to collect feedback in many different Case Types.

Processes can include Pega GenAI™ features to further automate your Case processing, for example, to populate data fields with personalized, AI-generated

©2025 Pegasystems Inc

information. For more information, see Process shapes and Populating Case data with personalized AI-generated information.

For more information see Subprocesses and Calling one Process from another Process.

User actions

Assigns a predefined task to a user or Work Queue, such as adjusting a goal and deadline, or changing a Stage. Resolving this Step type requires a human action before a Case can continue.

For more information, see Configuring and working with optional Actions in Cases.

Automations

Provides preconfigured functionality, such as sending an email or creating a Case. Depending on the Automation, you can configure additional options for a Step. For example, for a Step that automatically sends emails, you can apply business logic to define the recipient and provide the subject line and content.

For more information about automations, see Adding an automated Step to a Process.

Form Steps

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

The following figure shows a run-time presentation of two Form Steps:

PEGA ©2025 Pegasystems Inc

Form Steps at run time

For more information, see Adding a Multi-step Form to a Stage.

- Case Lifecycle elements
- Process shapes

Collecting information from a user

Gather the information that your business process requires by creating an Assignment Step to collect that data from users. For example, a customer service representative can enter personal details and income information from a client in the Loan Request Process.

> **Before you begin:**
>
> Create fields that capture data. For more information, see Creating fields for capturing data.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.

©2025 Pegasystems Inc

2. On the **Workflow** tab, click **Lifecycle**.
3. In the **Case Lifecycle** section, hover over a Process in a Stage, and then click **Step** > **Collect information**.
4. In the text field, provide a unique name for the Step by replacing the default Step label with a descriptive name.

> **For example:**
>
> Enter `Collect personal details`.

> **NOTE:** As a best practice, use descriptive, business-oriented names for Steps, so that different stakeholders across your enterprise can understand the business value of each Step. Typically, naming Steps is within the responsibility of business representatives.

5. In the **Step** property pane, click the **General** tab.
6. In the **Route to** list, identify the user or Work Queue 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, by user reference, by the reporting manager of the user who last updated the Case, or to a Case participant.

   - To assign the task to users that share a Work Queue, select **Work Queue**, and then, in the list of values, select or enter a Work Queue name.

     > **NOTE:** When you create a new Work Queue by selecting **Define new Queue** and provide the new Work Queue name, the

©2025 Pegasystems Inc

> Organization information (Org, Div, and Unit) for the new Work Queue is copied from the application. The work group that is assigned to the new Work Queue is the same as the creating operator's current work group.

- To assign the task to a Worklist or a team that shares a Work Queue, select **Custom**, and then configure the team or Worklist parameters.
- To automatically assign the task to a user with the appropriate availability, skill set, or workload, select **Use business logic**, and then click **Configure business logic** to define the conditions.

7. Click **Configure View**, and then define the fields on the form that users see when they perform the task.

   For more information, see Creating form Views for Steps.

8. **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.

     For more information, see Case Status values.

   - To create a new status, enter a unique name for the status.

     For more information, see Defining a Case Status.

9. **Optional:** To describe the purpose of the task, in the **Instructions for user** field, enter additional information about the Step.

> **For example:**
>
> `Enter personal details.`

©2025 Pegasystems Inc

Result:

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

Instructions for an Assignment

10. **Optional:** To ensure a timely resolution of the Assignment, define the goal and deadline for the Assignment.

    For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

11. Click **Save**.
12. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

What to do next:

Start planning your Case Type by associating Personas with your Case Type. For more information, see Adding Personas to organize users.

©2025 Pegasystems Inc

- Assignment shapes in Processes
- Case Lifecycle elements

Assigning tasks to users

Improve business process efficiency by assigning tasks to appropriate individuals or groups of individuals. For example, you can create an Assignment for a customer service representative (CSR) to follow up with a customer, and then route that Assignment to the CSR Work Queue.

For relevant training materials, see the Routing Assignments to users module on Pega Academy.

Before you begin:

Create an Assignment that you want to route. For more information, see Adding Steps to Processes.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

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 Lifecycle** section, select an Assignment that you want to route.
4. In the **Route to** list, identify the user or Work Queue that performs the task:

©2025 Pegasystems Inc

- 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, by user reference, by the reporting manager of the user who last updated the Case, or to a Case participant.

- To assign the task to users that share a Work Queue, select **Work Queue**, and then, in the list of values, select or enter a Work Queue name.

> **NOTE:** When you create a new Work Queue by selecting **Define new Queue** and provide the new Work Queue name, the Organization information (Org, Div, and Unit) for the new Work Queue is copied from the application. The work group that is assigned to the new Work Queue is the same as the creating operator's current work group.

- To assign the task to a Worklist or a team that shares a Work Queue, select **Custom**, and then configure the team or Worklist parameters.
- To automatically assign the task to a user with the appropriate availability, skill set, or workload, select **Use business logic**, and then click **Configure business logic** to define the conditions.

The following image shows an example Step configuration for routing to a specific user:

©2025 Pegasystems Inc

Routing a task to a specific user

5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- **Assigning users automatically at run time with business logic**
- **Configuring custom business logic-based routing**
- **Configuring custom routing logic for Assignments**
- Case Lifecycle elements
- Setting Service-Level Agreements for Stages, Processes, and Steps
- Adding optional Actions to Cases
- Assignment shapes in Processes
- Intelligent routing

©2025 Pegasystems Inc

Assigning users automatically at run time with business logic

Process your business cases with flexibility and efficiency by using business logic to automatically determine which individual or group of individuals receives an Assignment in a Case. Instead of entering specific names of users and teams, you can provide Conditions that your application evaluates at run time to assign a task. For example, your application can evaluate which customer service representative (CSR) has the most appropriate skill set or workload to complete a 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 **Lifecycle**.
3. In the **Case Lifecycle** section, click an Assignment in a Process.
4. In the **Step** section of the properties pane, click the **General** tab.
5. In the **Route to** list, select **Use business logic**, and then click the **Configure business logic** icon.

Location of 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**.

©2025 Pegasystems Inc

Actions	Steps
	b. In the **Value** field, press the Down arrow key, and then select or enter the name of a Work Queue. **NOTE:** When you create a new Work Queue by selecting **Define new Queue** and provide the new Work Queue name, the Organization information (Org, Div, and Unit) for the new Work Queue is copied from the application. The work group that is assigned to the new Work Queue is the same as the creating operator's current work group.
Route the Assignment by using custom business logic	a. In the **Route to** list, select an option from the **Custom options** 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:

©2025 Pegasystems Inc

a. In the list of values to evaluate, select the name of a field in your Data Model or a When Condition.
b. In the comparator list, select a comparator.
c. In the value field, enter or select a value that your application compares with the field in the Data Model or the When Condition.

> **For example:**
>
> Enter `Case Status is equal to "New"`

d. **Optional:** To define more Conditions that apply to this routing method, click the **Add row** icon, and then repeat steps 7.a through 7.c.
e. If you add multiple Conditions, between the rows, select the **and** or **or** operator to define how to evaluate the Conditions.

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

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. **Optional:** To change the order in which your application evaluates the Conditions, define a new order by dragging the Conditions.
10. In the **Otherwise** section, define a routing method that you want to use when no Conditions return a true value by repeating step 6.
11. Click **Submit**.

©2025 Pegasystems Inc

Result:

At run time, an application evaluates the Conditions and automatically routes an Assignment to the correct user or Work Queue. If an application evaluates all Conditions as false, it routes an Assignment by using an alternative method that you provide in the **Otherwise** section.

- Assignment shapes in Processes
- Configuring custom routing logic for Assignments
- Intelligent routing

Configuring custom business logic-based routing

To process Cases quicker, ensure that Assignments are routed to the most appropriate workers by using custom APIs for business logic-based routing. You can choose from default APIs or add custom APIs to meet your unique business needs. You can also override APIs to modify lists of available users and Work Queues, so that the lists contain only relevant workers.

For example, in a field service application you create an Assignment with business logic that routes the service request to the user whose postal code matches the location of the customer who raised the request. At run time, the system compares customer's postal code with the Conditions configured in the business logic. The system then routes the Assignment to the user whose postal code meets the Condition.

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

Related concepts

- Intelligent routing

Related tasks

- Assigning users automatically at run time with business logic

©2025 Pegasystems Inc

Adding custom routing options in business logic-based routing

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

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

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

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

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

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Data Model** category, and then click **Data Transform**.
3. On the **Instances of Data Transform** tab, open the *pyGetRoutingAPIs* Data Transform.
4. On the **Definition** tab, set the *pyStandardValue* parameter as equal to the name of your API.
5. Set the *pyPromptValue* parameter as equal to the label of your API.
6. Configure the return of the API by setting the *pyAPIReturnType* parameter:
   - To route the Assignment to a worker, enter `"OPERATOR"`.
   - To route the Assignment to a Work Queue, enter `"WORKQUEUE"`.
7. Click **Save as**.
8. Click **Create and open**.
9. **Optional:** Customize the source of properties that are mapped with parameters:
   a. In the navigation pane of Dev Studio, click **Records**.
   b. Expand the **Technical** category, and then click **Activity**.

©2025 Pegasystems Inc

   c. On the **Instances of Activity** tab, open *pySetRuleKeys*.
   d. Set the *Param.RuleClass* parameter as equal to the class that you want to use as the source for the parameters of your custom API.
   e. Click **Save as**.
   f. Click **Create and open**.
10. **Optional:** Include custom UI elements in the **Parameters** section:
   a. In the navigation pane of Dev Studio, click **Records**.
   b. Expand the **User Interface** category, and then click **Section**.
   c. On the **Instances of Section** tab, open *pyRuleParamsUIExtension* of a class *Data-MO-Activity-Router*.
   d. Customize the section to meet your specific needs.
   e. Click **Save as**.
   f. Click **Create and open**.

- Assigning users automatically at run time with business logic

Modifying lists of users and Work Queues for business logic-based routing

Modify the list of users and the list of Work Queues that are available for business logic-based routing of Assignments so that the lists contain only relevant workers. Modify the lists to ensure that the Assignment is routed to an appropriate person or Work Queue. For example, you can filter the list of users, so that you can only select workers with specific privileges.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Technical** category, and then click **Activity**.
3. Open *pyGetOperatorsOrWorkQueues*.
4. Modify the activity to meet your specific needs, for example, add a step to view the list of users that have certain privileges:
   a. Click **Add a step**.
   b. Enter a label for the step.
   c. Click **When**, and then select **Enable Conditions before this action**.

©2025 Pegasystems Inc

   d. Define when criteria. For more information, see Activity form - Completing the Steps tab - Entering preconditions.
   e. Click **Submit**.
5. Click **Save as**.
6. Click **Create and open**.

Related tasks

- Assigning users automatically at run time with business logic

Configuring custom routing logic for Assignments

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

As a best practice, use business logic instead of an activity to route Assignments, because this method is more flexible and easier to maintain. For more information, see Assigning users automatically at run time with business logic.

1. In the navigation pane of Dev Studio, click **App**.
2. On the **Classes** tab, expand the Case Type where you want to edit the Assignment, and then click **Process** > **Flow**.
3. In the **Flow** form, on the **Diagram** tab, double-click an Assignment.
4. In the **Assignment properties** dialog box, in the **Routing** section, from the **Route to** list, select **Custom**.
5. In the **Assignment type** list, choose a type of destination for the Assignment:
   - To route the Assignment to a specific user or Worklist, select **Worklist**.
   - To route the Assignment to a team or Work Queue, select **Work Queue**.
6. Specify the router activity that sets the *AssignTo* parameter to the name of a Worklist or a Work Queue:
   - To use an existing activity, in the **Router** field, press the Down arrow key, and then select the router activity.

©2025 Pegasystems Inc

- 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**.

- Automated and intelligent work assignment with Get Next Work
- Assignment shapes in Processes
- Intelligent routing

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.

©2025 Pegasystems Inc

Before you begin:

Add a Step to your Case Lifecycle. For more information, see Adding Steps to Processes.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

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 Lifecycle** section, click the Step that you want to supplement with instructions, for example, **Collect information**.
3. In the **Step** property pane, click the **General** tab.
4. In the **Instructions for user** field, enter additional information about the Step.

   For example:

   `Enter personal details`

5. Click **Save**.

   For example:

   The following figure shows the process of adding an `Enter personal details` instruction to an Assignment:

©2025 Pegasystems Inc

Adding instructions to an Assignment

6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

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

©2025 Pegasystems Inc

Instructions for an Assignment

- Step types
- Collecting information from a user
- Processes in a Case Lifecycle

Adding a Multi-step Form to a Stage

Break a single Assignment that captures many fields into multiple focused and concise screens by adding a Multi-step Form to a Stage. You can add both a sequential Process and a parallel Process as a Multi-step Form.

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

For relevant training materials, see the Multi-step Forms topic on Pega Academy.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Workflow** tab, click **Lifecycle**.
3. In the **Case Lifecycle** section, in a Stage, click **More** > **Add Process** > **New Multi-step Form**.
4. In the text field that represents the Process, replace the default label with a descriptive name.
5. In the properties pane on the right, click **General**.
6. On the **General** tab, in the **Start Process** field, select 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.

©2025 Pegasystems Inc

7. In the **Route to** list, identify the user or Work Queue 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, by user reference, by the reporting manager of the user who last updated the Case, or to a Case participant.

   - To assign the task to users that share a Work Queue, select **Work Queue**, and then, in the list of values, select or enter a Work Queue name.

     > **NOTE:** When you create a new Work Queue by selecting **Define new Queue** and provide the new Work Queue name, the Organization information (Org, Div, and Unit) for the new Work Queue is copied from the application. The work group that is assigned to the new Work Queue is the same as the creating operator's current work group.

   - To assign the task to a Worklist or a team that shares a Work Queue, select **Custom**, and then configure the team or Worklist parameters.
   - To automatically assign the task to a user with the appropriate availability, skill set, or workload, select **Use business logic**, and then click **Configure business logic** to define the conditions.
8. In the **Template** section, select the navigation type in the Multi-step Form by clicking **Change**, and then choosing the navigation template.
   The default setting is horizontal navigation.
9. Add a Step to the Multi-step Form by clicking **Form Step**.
   For more information about configuring Steps, see Steps in a Case Lifecycle.
10. In the text field, replace the default Step label with a descriptive name.

©2025 Pegasystems Inc

11. Click **Save**.
12. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

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

Multi-step Form at run time

©2025 Pegasystems Inc

For example:

The following figure shows a sample configuration of a Multi-step Form. It has two steps in the Multi-step Form named `Address` and `Valuation`, and features vertical navigation at run time. The process is configured to always start, and route to the current user of the application.

Adding a Multi-step Form

- Conditionally starting a Process
- Harnesses
- Configuring complex Processes
- Creating a Screen Flow in Dev Studio

Configuring advanced options for a Multi-step Form

To ensure that you deliver an application that precisely meets your business requirements, define how a Process that is a Multi-step Form behaves by configuring advanced starting options. For example, you can set a status that a Case enters when a Process starts, and then define a user to complete the Process.

©2025 Pegasystems Inc

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

> **NOTE:** In Dev Studio, a Multi-step Form is called a Screen Flow.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, hover over a Multi-step Form in a Stage, and then click the **Configure Process** icon.
3. Click the **Open Process** link, and then, on the canvas with the flow diagram, double-click the **Start** shape.
4. Perform any number of the steps from 3 to 11 in Creating a Screen Flow in Dev Studio.

Requesting approval from users

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

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

For relevant training materials, see the Designing an approval process module on Pega Academy.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.

©2025 Pegasystems Inc

2. On the **Workflow** tab, click **Life Cycle**.
3. In the **Case Lifecycle** section, in a Stage where you want to add an approval Step, click **Step** > **Approve/Reject**.
4. In the text field, provide a unique name for the Step by replacing the default Step label with a descriptive name.
5. In the **Step** properties pane, on the **General** tab, route the approval task:
   - To route the approval Step to a user in your application, in the **Route to** list, select **Specific user**, and then configure the user parameters.

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

     > **NOTE:** You can route the approval Step only to participants that have an account in your application.

   - To route the approval Step to users that share a Work Queue, in the **Route to** list, select **Work Queue**, and then select or enter a Work Queue name.

     > **NOTE:** When you create a new Work Queue by selecting **Define new Queue** and provide the new Work Queue name, the Organization information (Org, Div, and Unit) for the new Work Queue is copied from the application. The work group that is assigned to the new Work Queue is the same as the creating operator's current work group.

   - To route the approval Step to a user that your application assigns at run time based on Conditions that you define, select **Use business logic**, and then click **Configure business logic** to define the Conditions.

©2025 Pegasystems Inc

For example, an application can route an Assignment to a user with the most appropriate skill set and availability. For more information, see Assigning users automatically at run time with business logic.

Enable approval from

6. **Optional:** To enable approval by email and push notifications, in the **Enable approval from** field, select an approval method, and then perform steps 5 through 7 in Requesting approval by email and push notifications.
7. Click **Configure View**, and then define the fields on the form that users see when they perform the task.

   For more information, see Creating Views for Case Types.

8. 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 approves the Case.
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

©2025 Pegasystems Inc

Choices	Actions
	assign to the Case when a user approves the Case.

9. Define what happens when a user rejects the Case:

Choices	Steps
Reject and move 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.

10. Click **Save**.

©2025 Pegasystems Inc

Result:

Adding an Approval Step automatically creates an Alternate Stage in a Case Life Cycle. For more information, see Creating an Alternate Stage.

11. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

What to do next:

Extend the approval functionality:Improve the turn-around time for your requests by defining a goal and deadline for the approval Step. For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

- **Requesting approval by email and push notifications**
- **Configuring cascading approvals**
- **Configuring approvals with an authority matrix**

- Requesting approval by email and push notifications
- Setting Service-Level Agreements for Stages, Processes, and Steps
- Creating Views for Case Types
- Case Lifecycle elements

PEGA ©2025 Pegasystems Inc

Requesting approval by email and push notifications

Configure approval by email and push notifications to provide greater flexibility and convenience for users of your application when they process a Case. Users can then accept, reject, or take other actions on a Case without logging in to the application.

For example, a Case reaches an approval Step a the Lifecycle that sends an email with an approval request. The 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 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 on approval or moves the Case to an Alternate Stage in the Lifecycle on rejection.

Before you begin:

- Add an approval Step to your Case Lifecycle. For more information, see Requesting approval from a user.
- Manually create an email listener and a Service Email Rule. For more information, see Creating an email integration and Creating a Service Email Rule.
- To send approval requests by push notification, install your mobile app on a device, and then enable push notifications in the app. For more information, see Developing mobile apps for the traditional UI, Engaging mobile users with push notifications, Generating native mobile apps, and Configuring a mobile notification delivery Channel.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

©2025 Pegasystems Inc

For more information, see Creating a new application from a Blueprint.

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

   You can select both email and push notification forms.
5. In the **Email subject / Push notification message** field, enter text that you want to use as an email subject or a push notification message.
   You can reference property names in the subject to make it more dynamic and meaningful.

   For example:

   Enter `Case .pyCaseID is waiting for your approval.`

   Result:

   At run time, the application autopopulates the properties with relevant data. For example, the email subject is `Case LoanRequest-123 is waiting for your approval.`
6. If you request approval by email, define your message content:

©2025 Pegasystems Inc

Choices	Actions
Compose a message	a. In the Email content list, select Custom, and then click Compose email content. b. In the **Compose message** window, create your message by using the rich text editor. c. **Optional:** To add the email message to a template library to reuse in the future, click Actions > Add to approval template library, in the Template name field provide the template name, and then click Save. d. Click Done.
Reuse a template	a. In the Email content list, select Use existing. b. In the list of email templates, select a template that you want to use. You can select templates from your template library as well as from the default templates. c. **Optional:** To modify the message to meet your unique business needs, click Compose email content, modify the message by using the rich text editor, and then click Done.

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

©2025 Pegasystems Inc

Actions	Choices
Include all Case attachments	Select All Case attachments.
Include only specific Case attachments	a. Select Choose attachments. b. Click Add attachment. c. In the drop-down list, select whether you want to include an attachment of a particular category, or add the content of a field in your Case. d. In the **Select** field, select an attachment category or a field that you want to include as an attachment. **For example:** If you select the **Scanned Document** category, your application attaches all scanned documents to the email. **For example:** If you select a field that stores the amount of the loan requested, your application creates an attachment that contains the amount. e. To add more attachments, repeat steps 7.b through 7.d.

©2025 Pegasystems Inc

8. Click **Save**.
9. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

When the Case Lifecycle reaches the approval Step, your application sends an email or a push notification. A user who receives the approval request can take action on the Case without logging in to the application.

For example:

The following figure shows a sample configuration of an Approval/Rejection Step. At run time, a user to whom an application routes the approval receives an email notification. The email subject and the push notification include a Case ID and information that the Case awaits an approval. Also, the email includes all Case attachments.

©2025 Pegasystems Inc

Requesting approval by email notification

- Requesting approval from users
- Case Lifecycle elements

Configuring cascading approvals

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

For relevant training materials, see the Cascading approvals module on Pega Academy.

Before you begin:

Add an approval Step to your Case Lifecycle. For more information, see Requesting approval from users.

©2025 Pegasystems Inc

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, click the approval Step that you want to edit.
3. In the **Step** properties pane, on the **General** tab, in the **Approval flow type** list, select **Cascading**.
4. In the **Approval based on** list, select **Reporting structure**.
5. In the **Approval to be completed by** list, specify the user who is required to give their approval:
   - To assign the approval to the reporting manager of the user who processes the Case, select **Reporting manager**.
   - To assign the approval to a manager of the work group in which the user who processes the Case belongs, select **Workgroup manager**.
6. Specify how many levels of approval a Case requires:

Choices	Actions
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	a. Expand the **Approval level** section, and then select **Custom**. b. Click **Update custom levels**.

©2025 Pegasystems Inc

Choices	Actions
	c. In the window that appears, click **Add custom approval**. d. In the **When** field, enter a When Condition. The When Condition returns a Boolean expression that determines the manager required to approve the Case. e. In the **Levels of approval** field, enter how many levels of approval a Case requires. f. **Optional:** To add more conditions, repeat steps 6.c through 6.e. g. Click **Submit**.

7. In the **Step** properties pane, 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 their approval.
Approve and change Stage	a. In the **If approved then** list, select **Change Stage**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **To** list, select the Stage that you want to assign to the Case when a user gives their 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 their approval.

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

Choices	Steps
Reject and move Case to the next Step	a. In the **If approved 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**.

©2025 Pegasystems Inc

Choices	Steps
	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.

9. Click **Save**.
10. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **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 Requesting approval by email and push notifications.

- 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.

For relevant training materials, see the Cascading approvals module on Pega Academy.

PEGA ©2025 Pegasystems Inc

Before you begin:

- Add an approval Step to your Case Lifecycle. 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.
- Create a decision table that defines the users required to give their approval. For more information, see Creating decision tables.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, click the approval Step that you want to edit.
3. In the **Step** properties pane, on the **General** tab, in the **Approval flow type** list, select **Cascading**.
4. In the **Approval based on** list, select **Authority matrix**.
5. Define the users required to give their approval in the Step:
   a. **Optional:** In the **Decision table for matrix** field, enter a decision table that determines who is required to give their approval in the Step.
   Decision table populates the page list property. Number of the entries in the page list property defines the number of users required to give their approval. The approver property that you include in the decision table, holds the approver information in the page list property.

©2025 Pegasystems Inc

b. In the **Page list property** field, enter a page list that stores the users giving their approval.
c. In the **Approver property** field, enter a property that references the approving users.

6. In the **Step** properties pane, 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 their 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 their 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 their approval.

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

Choices	Steps
Reject and move Case to the next Step	a. In the **If rejected then** list, select **Continue**.

©2025 Pegasystems Inc

Choices	Steps
	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**.
9. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

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 Requesting approval by email and push notifications.

- Completing work on time
- Sending event notifications from Cases

Adding an automated Step to a Process

Save time and design more efficient business processes by adding preconfigured Steps to your Case Types. When you include a preconfigured Step in your Case Lifecycle, you provide an Automation that an application performs to bring a Case closer to a resolution.

For example, you can include a Step that automatically sends a notification to a manager when a Case awaits approval.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

For greater flexibility and efficiency, Pega Platform offers a wide variety of Automations that you can incorporate into your Case Lifecycle. To ensure that your application meets all of your business requirements, each Step provides additional configuration. For example, for a preconfigured Step that sends a notification, you can configure how your application determines the recipient and define the notification content.

PEGA ©2025 Pegasystems Inc

> **NOTE:** As a best practice, use descriptive, business-oriented names for Steps, so that different stakeholders across your enterprise can understand the business value of each Step. Typically, naming Steps is within the responsibility of business representatives.

You can add the following out-of-the-box Automations to your Case Lifecycle:

- **Attaching content to a Case**
- **Creating a PDF from Case data**
- **Starting child and Remote Cases from the main Case context**
- **Creating contextual Cases in traditional UI applications**
- **Collecting digital signatures in a Case**
- **Finding duplicate Cases**
- **Generating Case documents**
- **Automatically moving a Case to a different Stage**
- **Pausing and resuming Processes in Cases**
- **Populating Case data with a Data Transform**
- **Populating Case data with personalized AI-generated information**
- **Posting messages to a Case**
- **Preloading a Data Page**
- **Propagating data between parent and child Cases**
- **Running a queued background process in a Case**
- **Running a questionnaire in a Case**

PEGA ©2025 Pegasystems Inc

- **Sending event notifications from Cases**
- **Sending automatic emails from Cases**
- **Sending push notifications from Cases**
- **Saving data in a Data Page as part of a Case Lifecycle**
- **Spinning-off independent Cases from the current Case**

- Case Lifecycle elements
- Creating a Microjourney for customer success
- Setting Service-Level Agreements for Stages, Processes, and Steps
- Defining an escalation Action for an incomplete Assignment

Attaching content to a Case

Facilitate business interactions and provide users with the supporting information for a Case by attaching relevant correspondence and documentation to a Case Type. Use the Attach content Step to control the kind of information that Case workers add to a Case.

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

> **NOTE:** If you plan to customize attachment functionality, you should account for Basic Access Control. Register activities properly in order to avoid security issues going forward. For more information, see Basic Access Control.
>
> For relevant training materials, see the Access Control module on Pega Academy.

1. Add the Attach content Step to the workflow of your Case Type:

©2025 Pegasystems Inc

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 **Lifecycle** tab, in n the Stage in which you want users to be able to attach content, click **Step** > **More** > **Automations** > **Attach content**.

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

c. Click **Select**.

2. In the **Step** properties pane, select and configure the attachment type:

Choices	Actions
Attach a document	a. Click **Attach a file**. b. Click **Upload**. c. In the **Upload a file** window, click **Choose File**, and then select the file to upload. d. Click **Done**. e. In the **Select file** list, select the file to attach. f. In the **Attachment name** field, enter a descriptive name for the file. g. **Optional:** To change the business classification of the attachment, select an option in the **Attachment category** field.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: You can add attachment categories, which group attachments according to their business classification, on the Settings tab, in the **Attachment categories** section. For example, you can add the Proof of identity and Proof of address attachment categories that group two different sets of documents required in processing a request for a service offered by the government.
Attach a URL	a. Click Attach a link. b. In the URL field, enter a URL that points to an external file or resource on the Internet. c. In the URL name field, enter a descriptive name for the link. d. **Optional:** To change the business classification of the attachment, select an option in the Attachment category field.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: You can add attachment categories, which group attachments according to their business classification, in the **Attachment categories** section, on the Settings tab. For example, you can add the Proof of identity and Proof of address attachment categories that group two different sets of documents required in processing a request for a service offered by the government.
Attach a note (available only in Dev Studio) **NOTE: This option is for backward compatibility only and is not recommended. Instead, use Pulse to add notes and collaborate on Cases.**	a. In the navigation pane of Dev Studio, click Cases, and then click the Case Type that you want to open. b. Perform steps 1b through 1c. c. In the Step properties pane, click Attach a note. d. In the **Note** text field, provide a note that you want to attach.

©2025 Pegasystems Inc

Choices	Actions
For more information, see Configuring Pulse for Case Types.	e. In the **Attachment description** field, enter a descriptive name for the note.

3. **Optional:** To provide more information for auditing purposes, in the **Audit note** field, enter additional information.
4. Click **Save**.
5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

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

For example:

The following figure shows a design-time configuration of an Attach content Step and the run-time effect. Users that process a Case can view and download the `Certificate of employment` document that an application automatically attaches to a Case.

©2025 Pegasystems Inc

Content attached to a Case

What to do next:

You can also enable users to attach files to cases at run time by configuring the Attach content control. For more information, see Configuring the Attach content control.

- Controlling access to Case attachments
- Process shapes
- Adding an automated Step to a Process

Creating a PDF from Case data

Document the values that users enter during Case processing by creating screen captures of user inputs and then saving them as PDF files. By documenting input values, you can understand how users interact with a Case and, after Case resolution, analyze the information that users provide. For example, you can create a PDF that contains details of a customer complaint so that you can download the PDF and attach it to relevant offline documentation.

©2025 Pegasystems Inc

Before you begin:

For Constellation applications, use traditional Section Rules and HTML Rules to generate documents. For more information, see Generating PDF in Constellation.

NOTE: PDF is a restrictive format that forces hard constraints on settings such as margins and layouts. Consequently, the *HTMLToPDF* Rule does not support the conversion of dynamic layout groups. As a best practice, use free-form layouts or smart layouts with forms that you want to print to PDF.

Adding the Create PDF Step to the Case Lifecycle:

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
   a. In a Process in which you want to add the Step, click **Step** > **More** > **Automations** > **Create PDF**.

      NOTE: If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

   b. Click **Select**.

Configuring the automated Step:

2. In the **Step** properties pane, select the data from which you want to create a PDF:

©2025 Pegasystems Inc

Choices	Actions
Section	a. In the **Generate PDF from** list, select **Section**. b. In the **Select View** list, select a View from which you want to create a PDF capture. **For example:** To create a PDF file that contains Case attachments, select **Case attachments**. c. In the **PDF orientation** section, specify the file layout. d. In the **PDF name** field, enter a descriptive name for the document, for example, `Customer Complaint`. e. In the **Attachment category** list, select an option that describes the business classification of the capture. f. **Optional:** To include a note about the information, in the **Audit note** field, enter some meaningful text.
eForm template	a. In the **Generate PDF from** drop-down menu, select **eForm template**. b. In the **eForm template** field, enter the map rule reference for the map from which you want to create a PDF capture.

©2025 Pegasystems Inc

Choices	Actions
	c. In the **PDF file name** field, enter a descriptive name for the document, for example, `Complaint`. d. In the **Attachment category** list, select an option that describes the business classification of the capture. e. **Optional:** In the **Audit note** field, enter an audit note.

3. Click **Save**.
4. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

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

For example:

The following figure shows a form that customers complete to file a complaint about their stay at a hotel. At run time, the form is available as a PDF for the customer service representatives (CSRs) who process the Case. CSRs can then view and download the PDF that contains the information that the customer provides.

©2025 Pegasystems Inc

PDF attached to a Case

- Attachment types
- Restricting user actions for Case attachments
- Process shapes
- Adding an automated Step to a Process

Starting child and Remote Cases from the main Case context

Deliver more granular and flexible applications by creating a contextual Case that starts when your current Case is in progress. Instead of defining one long, complex Case Type, you can trigger a contextual Case at the appropriate point in your main Case Type.

©2025 Pegasystems Inc

For example, in an application to review insurance claims after car accidents, you can start a contextual Case to collect and review materials so that a customer service representative can estimate the damage before returning to the main Case.

You create contextual Cases with a Create Case Step, which you can configure to run a child Case or a Remote Case from the main Case context.

For more information about these two types of Cases, see Case Types and Remote Case Types.

Before you begin:

- Ensure that your application uses Constellation. For more information, see Application UI setup.
- If you want to start a Remote Case from your current Case Type, ensure that you have at least one Remote Case Type configured in your application. For more information, see Creating Remote Case Types.
- For the Remote Case contextual Case scenario, ensure that you create a property in the Data Model of your Case Type for referencing Remote Case Types by the Case IDs.

  For more information about Case reference fields, see Referencing a Case Type.

  For more information about query fields, see Querying a Data Page.

> **NOTE:** When you create a Remote Case Type, the system creates two Data Pages that you can use in your Query field to fetch a list and the *pyIDs* of Cases of this Case Type.
>
> For example, when you create a `Review loan request` Remote Case Type, the system creates the `D_ReviewLoanRequestList` and `D_ReviewLoanRequest` Data Pages.

©2025 Pegasystems Inc

> You can also create your own query field that fetches a list of Cases based on your parameters, for example, a specific value of the *ContactID* parameter. For more information, see Fetching a list of selected Remote Cases with a queryable Data Page.

1. Add the Create Case Step to the Lifecycle of your Case:
   a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
   b. In the Process in which you want to add the **Create Case** Step, click **Step** > **More** > **Automations** > **Create Case**.

   > **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

   c. Click **Select**.
   d. In the **Create Case** field, enter the name of the contextual Case Type, for example `Damage estimates`.
2. In the **Step** properties pane, in the **Create** field, define the Case Type that you want to use:

Choices	Actions
Start a child Case	a. Select **Child Case**. **NOTE:** If you create a child Case Type, you can complete your main Case

©2025 Pegasystems Inc

Choices	Actions
	Type only after the child Case Type reaches resolution. b. In the **Case Type** field, select or create the Case Type that you want to use. c. **Optional:** To create more child Case Types, select the **Create multiple Cases using a list** checkbox, and then, in the **List field** list, select a field group that holds details of Cases to create at run time. The value is sourced from a query field that you configure to store and collect the list of Case Types. For more information, see Querying a Data Page and Configuring query fields. d. **Optional:** To reuse the parent Case data in the child Case, select the **Transfer information to new Case** checkbox, and then, in the **Transfer information** dialog box, select the fields that you want to map to the child Case Type by performing steps 6.a through 6.d in Creating a child Case Type in Case Designer.
Start a Remote Case	a. Select **Remote Case**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **Case Type** field, select the Remote Case Type that you want to use. c. **Optional:** To reuse the parent Case data in the Remote Case, select the **Transfer information to new Case** checkbox, and then, in the **Transfer information** dialog box, select the fields that you want to map to the Remote Case Type by performing steps 6.a through 6.b, and 6.d in Creating a child Case Type in Case Designer. **NOTE:** Ensure that the **To** fields that you map are configured in the remote application. For more information, see Adding fields while creating cases in Constellation DX API . d. In the **Field to store Remote Case reference** list, select a Case reference field that stores the ID of the Remote Case. **NOTE:** The list displays only Case reference fields that use Data Pages that

©2025 Pegasystems Inc

Choices	Actions
	have the alternate key storage parameter enabled and have `pyID` selected in the **Linked field**. For more information, see Adding alternate keys for external reference fields and step 7 in Referencing a Case Type. When you create a Remote Case Type, the system creates two Data Pages that fetch a list and the *pyIDs* of Cases of this Case Type. For example, when you create a `Review loan request` Case Type, the system creates the `D_ReviewLoanRequestList` and `D_ReviewLoanRequest` Data Pages. For more information, see Referencing a Case Type and Querying a Data Page. During Remote Case Type registration, the system creates an autogenerated Page-structured Data Page named `D_`*[Remote Case Type name]*, which has the **Is this page used for alternate key storage?**

©2025 Pegasystems Inc

Choices	Actions
	checkbox selected and the **Linked field** set as `pyID`. **NOTE:** During Remote Case Type creation, if a Page-structured Data Page with the same name already exists in the application, the Data Page configuration is not updated, which means that the **Is this page used for alternate key storage?** checkbox is not selected. You might want to use a custom query field to fetch a filtered list of Cases based on your parameters. For more information about query fields, see Fetching a list of selected Remote Cases with a queryable Data Page.

3. **Optional:** To provide more information for auditing reasons, in the **Audit note** field, enter an audit note.
4. **Optional:** To enable users to create more contextual Cases from the current Case context, repeat steps 1 through 3.
5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

©2025 Pegasystems Inc

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

The following figure shows a run-time display of the Damage estimates contextual child Case that you run from the Insurance Claims Review Case Type:

Contextual child Case in a parent Case Type View

The default name of the tab follows the *[child Case Type name]* **list** pattern, which you can later change by performing step 5 in Configuring a Full Page View .

What to do next:

If you set up Remote Case creation, configure a View to display a tab and a list of Remote Cases in the parent Case context. For easy setup of the View, you can select the *D_pxRemoteCasesByCase* Data Page to view all the Remote Cases created from the parent Case. For more information, see Configuring a Full Page View.

©2025 Pegasystems Inc

> **NOTE:** In the child Case option, the system automatically adds a list View with a list of related child Cases in the parent Case context.

Related concepts

- Accessing Remote Cases in the consumer application
- Case Types
- Remote Case Types
- Using Data Pages in a Case Lifecycle

Related tasks

- Extending a Data Model for a Case
- Creating fields for capturing data

Creating contextual Cases in traditional UI applications

To provide additional processing in your business processes, create a contextual Case that starts when your current Case is in progress. Creating contextual Cases helps you deliver more granular and flexible applications. Instead of defining one long, complex Case Type, you can trigger a contextual Case at an appropriate point in your main Case Type.

For example, in an application to review insurance claims after car accidents, you can start a contextual Case to collect and review materials so that a customer service representative can estimate the damage before returning to the main Case.

> **NOTE:**
>
> - As a best practice, create workflows with Pega GenAI Blueprint™. For more information, see Creating a new application from a Blueprint and Pega GenAI Blueprint - doc journey.

©2025 Pegasystems Inc

- Creating contextual Cases in Dev Studio is suitable for advanced developers and typically involves creating a Data Transform to propagate data between the two Case Types. As a best practice, take advantage of low-code solutions in App Studio. For more information, see Creating a child Case Type in Case Designer and App Studio overview.
- Additionally, for complex workflows and asset reuse, consider evaluating alternative approaches to determine the best fit for your business scenario:
  - Spin-off Cases: This technique initiates Subprocesses that run in parallel with the main process without waiting for their completion, which is suitable for tasks that can proceed independently.

    For more information, see Subprocesses.

  - Parallel processing: Implementing parallel Processes within the same Case can enhance efficiency by allowing multiple tasks to proceed simultaneously.

    For more information, see Adding a parallel Process to a Stage

1. Add the **Create Case** Step to the lifecycle of your main Case Type:
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. In the Process in which you want to add the **Create Case** Step, click **Step** > **More** > **Automations** > **Create Case**.

   NOTE: If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

©2025 Pegasystems Inc

c. Click **Select**.
d. In the **Create Case** field, enter the name of the new Case that you want to start from the context of the main Case, for example `Damage estimates`.

2. In the **Step** properties pane, in the **Create** field, specify the type of the Case that you want to start:
   - To start an independent Case, select **Case**.
   - To start a child Case of the current Case, select **Child Case**.

> **NOTE:** If you create a child Case, you can complete your main Case only after the child Case reaches resolution.

3. Define a Case Type to embed in the context of the main Case Type:

Choices	Actions
Create a new Case Type	a. In the **Create the following** Case list, select **Create new Case Type**. b. In the **Case Type name** field, enter the name of your Case Type. c. Click **Create**.
Use an existing Case Type	In the **Create the following Case** list, select the Case Type that you want to use.
Use an existing Case Type by selecting a class	a. In the **Create the following Case** list, select **Other**. b. In the **class** field, press the Down arrow key, and then select the class that stores the Case Type that you want to use. When you select a class, you do not need to provide a run-time Case Type name.

©2025 Pegasystems Inc

4. If you create a child Case Type, to create more than one Case Type, select the **Create multiple Cases using a list** checkbox, and then, in the **List field** field, press the Down arrow key, and then select a field group that determines the number of Cases to create.
5. **Optional:** To propagate data from your current Case to the new Case, in the **Data Transform** field, press the Down arrow key, and then select the name of a Data Transform.
   For more information, see Data Transforms and Configuring a Data Transform.
6. If you create a new Case Type, to obtain a reference to the new Case that you can use in the current Case, in the **Property to store ID of Case** field, press the Down arrow key, and then select the name of a Single-value property.
7. **Optional:** To provide more information for auditing reasons, in the **Audit note** field, enter the audit note.
8. Click **Save**.

- Case Types
- Steps in a Case Lifecycle
- Creating a Case Type Rule form in Dev Studio

Collecting digital signatures in a Case

Enhance your business transactions with relevant information by electronically signing, reviewing, and sharing documents. When you integrate your Case Types with DocuSign, you can collect digital signatures for your Case attachments, and as a result, automate the review process for documents and resolve Cases more quickly.

When you collect digital signatures, you avoid producing and exchanging printed documents, which might take longer. For example, in a hiring process, a hiring manager might digitally sign a decision to approve a job candidate.

PEGA ©2025 Pegasystems Inc

Before you begin:

Create and configure a DocuSign account. For more information, see Integrating DocuSign with Pega Platform.

1. Add the Send via DocuSign Step to the Lifecycle of your Case:
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. In the Process in which you want to add the Send via DocuSign Step, click **Step** > **More** > **Automations** > **Send via DocuSign**.

   > **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

   c. Click **Select**.
2. In the **Step** properties pane, in the **Envelope** field, enter the envelope that you want to use.
   A DocuSign envelope stores information that is necessary to make a successful signature request to DocuSign.
3. **Optional:** To provide more information about this Step for auditing purposes, in the **Audit note** field, enter the information that you want to give.
4. **Optional:** To allow users to navigate to this Step from the bread crumb navigation at run time, select the **Enable navigation link** checkbox, and then specify additional options:
   a. **Optional:** To allow users to navigate to the Step only after this Step is complete, select the **Only allow navigating back to this Step** checkbox.

©2025 Pegasystems Inc

   b. **Optional:** To enable post-processing or validation when users navigate back from the Step, select the **Perform post-processing when navigating away from Step** checkbox.
5. Click **Save**.
6. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Related tasks

- Configuring a signature request to DocuSign
- Attaching content to a 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. For example, you can check if a customer creates a loan request for a specific amount of money before a previous loan request from the same customer reaches resolution.

Duplicate Cases are Cases that have the same data and purpose. For example, a customer can create two orders that have the same products and quantities. The application must evaluate whether the second order is a duplicate, and if it is, flag it accordingly. In Pega Platform™ you can use an out-of-the-box Automation that helps detect duplicate Cases, which you add to the Case Lifecycle as a Step, and then, in the Step, you configure the required as well as the fine-tuned Conditions for data comparison between Cases:

PEGA ©2025 Pegasystems Inc

Basic conditions
: The required Conditions that must match for a Case to be considered duplicate.

Weighted conditions
: If the basic Conditions are met, the system analyzes the additional, weighted Conditions that can be assigned a score and threshold, and can also be defined as the sum of other weighted Conditions.

The basic and weighted Conditions compare property values in the application to determine whether a Case is a duplicate.

A Case is deemed duplicate when the basic Conditions are matched and the sum of the weighted Conditions matches or exceeds the threshold value.

- Adding an automated Step to a Process

Searching duplicate Cases

Save time and minimize creation of similar Cases by adding a Search duplicate Cases Step to your Case Lifecycle.

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 **Lifecycle**.
3. In a Process to which you want to add a Search duplicate Cases shape, click **Step** > **More** > **Automations** > **Search duplicate Cases**.

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

4. Click **Select**.

©2025 Pegasystems Inc

Result:

Your Case Type now includes a Search duplicate Cases shape, as shown in the following figure.

Search duplicate Cases Step

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 from the navigation pane of a Portal
- Identifying duplicate cases

Defining basic Conditions for Case Types

Improve the performance of your duplicate Case search and limit the number of potential duplicates by defining basic Conditions for potential duplicate Cases.

©2025 Pegasystems Inc

Before you begin:

Add a Search duplicate Cases Step to the Lifecycle of your Case Type. For more information, see Searching duplicate Cases.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the Case working area, on the **Workflow** tab, click **Lifecycle**.
3. In the **Case Lifecycle** section, click the **Search duplicate Cases** Step for which you want to define basic Conditions.
4. In the **Step** property pane, click **Add basic Condition**.
5. In the dialog box that appears, in the **Potential duplicate**, **Comparator**, and **Current Case** fields, enter the property names and values that you want to use to determine duplicates in a comparison against the existing Cases.
   The application uses these values to compare existing Cases against the current Case to identify potential duplicates.

   For example:

   To compare your Case with all Cases that contain the same email address, enter `Email is same`.

   NOTE: You need to expose the properties that are referenced by an exact Condition as columns in the database. For more information, see Optimizing database properties.

6. **Optional:** To define more basic Conditions, in the dialog box, click **Add a row**, and then go to step 5.
7. **Optional:** To mark the Case as a potential duplicate only when the Case meets all basic Conditions, in the field below **When**, select **and**.

©2025 Pegasystems Inc

8. **Optional:** To mark the Case as a potential duplicate when the Case meets at least one basic Condition, in the field below **When**, select **or**.
9. Click **OK**.
10. Click **Save**.
11. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

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.

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

In the results table, you can preview the matching property values of a potential duplicate Case and the total score contribution number by clicking **Why is this shown here?**. A dialog box is displayed that provides the reason for marking the Case as a potential duplicate, which helps you make an informed decision about what action to take.

> **NOTE:** Only simple match scores are shown in the dialog box. The *Date time* and *Date* property types are complex match scores and do not display the values. However, the total score contribution of complex matches is shown in the **Other Conditions** section at the bottom of the

©2025 Pegasystems Inc

> dialog box, so the lack of display has no impact on duplicate Case evaluation.

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 an automated 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 Step to the Lifecycle of your Case Type. For more information, see Searching duplicate Cases.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the Case working area, on the **Workflow** tab, click **Lifecycle**.
3. In the **Case Lifecycle** section, click the **Search duplicate Cases** Step for which you want to define the weighted Conditions.
4. In the **Step** property pane, click **Add weighted Condition**.
5. In the dialog box that appears, in the **Weight** field, enter a value between `1` and `100` that you want to add to the total weight of a Case when the Case meets the Condition.

©2025 Pegasystems Inc

For example:

When you create three Conditions with the weight of `20` each, the total weight is `60`.

NOTE: If you want the application to display additional information in a notification about duplicate Cases, include the properties that contain that information as columns in a version of the *pyCaseMatchProcessingTemplate* report definition that applies to the class of your Case Type. For more information, see Optimizing database properties.

6. In the **Potential duplicate**, **Comparator**, and **Current Case** fields, enter the property names and values that you want to use to determine duplicates in a comparison against the existing Cases.
   The application uses these values to compare existing Cases against the current Case to identify potential duplicates.

   For example:

   In a hotel booking application, to compare your Case with all Cases that contain the same room number, enter `Room number is equal to 1006`.

7. Click **OK**.
8. **Optional:** To add more weighted Conditions, repeat steps 4 through 7.
9. In the **And weighted Conditions sum at least** field, enter a sum of weights that represents a threshold when the Case is a potential duplicate.

TIP: The threshold value should be lower than the sum of all weighted Conditions.

©2025 Pegasystems Inc

10. Click **Save**.
11. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

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

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

For example:

The following figure shows a sample configuration for finding duplicate Cases at a hotel booking application. The configuration includes a basic Condition that searches for Cases with the same email, and three weighted Conditions that cover dates of arrival and a departure, and a room number. At run time, an

PEGA ©2025 Pegasystems Inc

application marks a Case as a duplicate when the email is the same as in a Case that already exists, and then weighted Conditions that evaluate to true have total value of at least 50.

Conditions for duplicate Cases

The following video shows how to add a Search duplicate Cases Step to a Case Life Cycle, and then define basic and weighted Conditions:

©2025 Pegasystems Inc

- Adding an automated Step to a Process

Defining Views for duplicate Cases search results

Ensure that your users have the information that they need to make a decision on duplicate Case resolution by configuring the fields and text that are displayed when the duplicate Cases search is complete.

> **Before you begin:**
>
> Add a Search duplicate Cases Step to the Lifecycle of your Case Type. For more information, see Searching duplicate Cases.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the Case working area, on the **Workflow** tab, click **Lifecycle**.
3. In the **Case Lifecycle** section, click the **Search duplicate Cases** Step for which you want to define the View.
4. In the **Step** property pane, click **Configure view**.

©2025 Pegasystems Inc

The configure View button in the Step property pane

5. In the **Edit View** dialog box, in the **Instructions** list, define the kind of additional guidance that you want to provide to your users:
   - To hide Step instructions, select **None**.
   - To define a custom message, select **Override Case Step instructions**, and then, in the **Instruction text** field, enter a custom Step instruction.
   - To display the instructions that are defined in the **Case Life Cycle**, select **Case Step instructions**.

   For more information, see Adding instructions to Assignments.
6. In the **Columns** section, add the columns that you want to include in the view by clicking **Add**, and then select the field that you want to associate with the column.
7. In the row with the newly added column, click the **Configure** icon, and then define additional settings such as labels and visibility.
   For more information, see Configuring fields in the UI.
8. Click **Submit**.
9. Click **Save**.

©2025 Pegasystems Inc

10. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

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 Lifecycle. You create Case documents by using templates to ensure that you deliver relevant 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 the Data Model of a Case Type
- 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:**
>
> - For Constellation applications, use traditional Section Rules and HTML Rules to generate documents. For more information, see Generating PDF in Constellation.

©2025 Pegasystems Inc

- 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 Extending 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** Step, click **Step** > **More** > **Automations** > **Generate document**.

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

3. In the **Step** properties pane, 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.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: To ensure full compatibility and correct rendering of dynamic fields such as hyperlinks, create your .docx template using a desktop word processor that fully supports .docx formatting. Templates created in online or browser-based editors may not be processed correctly by the document generation engine, even if saved in .docx format. For best results, open the generated documents in a word processor with full .docx compatibility. b. In the **Step** properties pane, 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.

©2025 Pegasystems Inc

Choices	Actions
	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.
7. **Optional:** To view how your changes are reflected at run time, click Preview and run the Case.

> **IMPORTANT:** Avoid using Save and run to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

Result:

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

For example:

The following figure shows a template with tags that correspond to a Case ID, an arrival date, and a departure date in a Case of filing a complaint at a hotel. The figure also includes a run-time effect with data specific to a Case that an application populates based on tags in the template.

Generating a Case document

- **Case documents tags**
- **Case documents specifications**
- **Case documents error messages**
- **Activities for integrating Cases with Excel documents**

©2025 Pegasystems Inc

- **Tags for generating .docx documents through APIs**

- Creating project documents for stakeholders
- Exporting a Data Model into a document

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","type":"P",setTZ="true"}>` Run time: `09/19/2019 11:30:15 AM GMT`
	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":"`

©2025 Pegasystems Inc

Type	When to customize	Configuration parameters	Example
			`.DateOfReport", "format":"MM/dd/yyyy","type":"P"}>` Run time: `09/19/2019`
Time only		"name":".PropertyName","type":"P"	Design time: `<pegaref config={"name":".TimeOfReport","format":"hh:mm a","type":"P","setTZ":"false"}>` Run time: `04:13 PM`
	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","isCurrency":"true`
	Display decimal places	"scale":"*Any number between 0 and 10*"	
	Group digits	"doGroupBy":"true", "groupBySize":"*Any number between 2 and 5*"	

©2025 Pegasystems Inc

Type	When to customize	Configuration parameters	Example

	Use symbol for currency	"isCurrency":"true","currency":"*Symbol*_val"	`",`
			`"currency":"$_val (USD)"}>`
			
			Run time: `$12,345,678.00 (USD)`

	Use code for currency	"isCurrency":"true","currency":"_val (*Code*)"	
	Use symbol and code for currency	"isCurrency":"true","currency":"*Symbol*_val (*Code*)"	
Boolean		"name":".PropertyName","type":"P","tValue":"ValueForTrue","fValue":"ValueForFalse"	Design time: `<pegaref config={"name":".PropertyName","type":"P","tvalue":"Yes","fvalue":"No"}>` Run time: `Yes`
Attachment		"name":".AttachmentFieldProp.pxResults(1).pyAttachmentLink","type":"A"	Design time: `<pegaref config={"name":".AttachmentFieldProp.pxResults(1).pyAttachmentLink","type":"A","showLabel":"true","attach`

©2025 Pegasystems Inc

Type	When to customize	Configuration parameters	Example
			`Label":".XYZ","width":"20","height":"30"}>` Run time: The application displays image attachments, and embeds files in a .docx format as inline attachments. Otherwise, the application adds attachments as objects by suing object linking and embedding (OLE).
Text		"name":".PropertyName","type":"P"	**NOTE:** Users can insert images of any size when they use a rich text editor. Design time: `<pegaref`
	Text area	"format":"TA"	
	Rich text editor	"format":"RTE"	

©2025 Pegasystems Inc

Type	When to customize	Configuration parameters	Example
			`config={"name":".PropertyName","type":"P","format":"TA"}>` Run time: Users can enter text in a text area.
Properties inside page list		"name":".PageList().PropertyName"	Design time: `<pegaref config={"mode":"pagelist","name":".PageList().PropertyName","type":"P"}>` Run time: Row 1: *Value1*, Row 2: *Value2*.
Properties inside page		"name":".Page.PropertyName"	**NOTE:** Document generation supports embedded page

©2025 Pegasystems Inc

Type	When to customize	Configuration parameters	Example
			levels, for example `.Page1.Page2.Page3.PropertyName`. Design time: `<pegaref config={"name":".Page.PropertyName","type":"P"}>` Run time: *PropertyValue*
Section		"name":"SectionName","type":"S"	**NOTE:** Document generation does not support skin styling and CSS3. As a

©2025 Pegasystems Inc

Type	When to customize	Configuration parameters	Example
			result, the application might not properly render Pega autogenerated sections. Design time: `<pegaref config={"name":"SectionName","type":"S"}>` Run time: *Section name*
Correspondence		"name":"CorrespondenceName","type": "C"	**NOTE:** Document generation does not support skin styling and CSS3.

©2025 Pegasystems Inc

Type	When to customize	Configuration parameters	Example
			As a result, the application might not properly render Pega autogenerated sections. Design time: `<pegaref config={"name":"CorrespondenceName","type":"C"}>` Run time: *Correspondence content*

- Generating Case documents

Case documents specifications

Ensure that your application generates the necessary documents in a Case Lifecycle 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

©2025 Pegasystems Inc

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.

PEGA ©2025 Pegasystems Inc

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

PEGA ©2025 Pegasystems Inc

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 and suggested resolutions:

Error message	Description	Resolution
`ERR_SKIP_HTML_IN_TABLE`	A table cannot contain content from a section, correspondence, a rich text editor, or HTML.	Replace the HTML or rich text content within the table with plain text or compatible elements. If necessary, move the unsupported content outside the table.
`ERR_SKIP_HTML_IN_HEADER/FOOTER`	A header or footer cannot contain content from a section, correspondence, a rich text editor, or HTML.	Redesign the header or footer to include only plain text or supported elements. Ensure that sections or correspondence are not referenced in these areas.
`ERR_RETRIEvING_HTML`	A document cannot retrieve HTML from a section or correspondence.	Check the section or correspondence configuration for errors. Verify that the source content is accessible and compatible with Case document generation. Correct any broken

©2025 Pegasystems Inc

Error message	Description	Resolution
		references or unsupported HTML tags.
`ERR_INVALID_CONFIGURATION`	The JSON configuration is invalid.	Validate the config JSON in the *pegaref* element.
`ERR_INVALID_TYPE`	The JSON type key is not a property, attachment, section, or correspondence.	Verify the type key in the config JSON. Update it to one of the supported values: property, attachment, Section, or Correspondence
`ERR_EMPTY_PROPERTY_NAME`	The JSON name key is empty.	Ensure that all required name keys in the config JSON are populated. If a name key is missing, identify the corresponding property or object and provide a valid name.
`ERR_SKIP_ATTACHMENT_IN_PARA`	A document cannot contain an attachment in a paragraph with text.	Move the attachment out of the paragraph or separate the text and attachment into distinct sections. Alternatively, add the attachment in a compatible area.
`ERR_INVALID_ATTACHMENT_INFO`	The JSON configuration is correct but the attachment keys are invalid.	Review the attachment keys in the config JSON. Ensure they match the keys defined in your application and

©2025 Pegasystems Inc

Error message	Description	Resolution
		correct any typos or mismatches.
`ERR_SKIP_NON_IMAGE_ATTACHMENT_IN_TABLE`	A table cannot contain non-image attachments.	Remove the non-image attachment from the table. If the attachment is required, convert it to an image format or place it in a compatible location outside the table.
`ERR_SKIP_NON_IMAGE_ATTACHMENT_IN_HEADER/FOOTER`	A header or footer cannot contain non-image attachments.	Replace the non-image attachment with a compatible format, such as an image, or remove it entirely from the header or footer.
`ERR_COULD_NOT_ADD_ATTACHMENT`	An application could not add an attachment to a document.	Review the attachment configuration for errors.
`ERR_ATTACHMENT_NOT_FOUND`	An application could not find an attachment that matches a provided key.	Ensure that the provided key is correct and that the corresponding attachment exists. Update the configuration or upload the missing attachment.
`ERR_ADDING_IMAGE`	An application failed to add an image to a document.	Verify if a valid image is referenced.
`ERR_PROCESSING`	An error occurred during processing of HTML or an attachment.	Review the HTML and attachment configuration for errors.

©2025 Pegasystems Inc

Error message	Description	Resolution
`ERR_PAGELIST_UNSUPPORTED`	A document does not support page lists outside tables.	Restrict page lists to tables only. Reorganize the document to ensure page lists are placed within table structures.

Additional tips for debugging

- Review logs: check the application logs for more detailed error descriptions and stack traces.
- Test incrementally: when debugging errors, test smaller sections of the document to isolate problematic content.

- Generating Case documents
- Properties
- System administration

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:

©2025 Pegasystems Inc

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 Pega 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 Pega	String	Yes	Yes	N/A
	Integer	Yes	Yes	N/A

©2025 Pegasystems Inc

Feature	Name	Supported in Pega Platform 8.1	Supported in Pega Platform 8.2 and later	Description
Clipboard data types	Double	Yes	Yes	N/A
	Date	Yes	Yes	N/A
	Boolean	Yes	Yes	N/A
Properties from any Pega 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: *{.pxResuls().value input}* , *{.description input}*
	PageList	Yes	Yes	
Styles	Font, cell, color	No	No	The application ignores cell styling and cell content styling while parsing data to the Pega Clipboard.

©2025 Pegasystems Inc

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
FSFileName	Yes	Yes	The name of the file to parse.
TemplateRFB	Yes	Yes	Template details in the *Fileformat*`!`*Filename*`!xlsx` format.
sheetsToParse	Yes	Yes	Sheet names stored as comma-separated values. If the parameter is empty, the application skips Validation and parses all of the sheets.
sheetWiseData	Yes	Yes	When checked, the application organizes parsed data sheetwise by using the *pySheets(sheet name)* page group.
hasHeader	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.
bDeleteFile	Yes	Yes	When the value is `true`, the application deletes the file after parsing.
outputPageName	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.

©2025 Pegasystems Inc

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.
Supported Excel	String	Yes	Yes	
	Numeric	Yes	Yes	

©2025 Pegasystems Inc

Feature	Name	Supported in Pega Platform 8.1	Supported in Pega Platform 8.2 and later	Description
cell types	Boolean	Yes	Yes	
	Formula	No	Yes	
	Date	No	Yes	
	Hyperlink	No	Yes	Sample cell value: *{.pxResults().linkToSheet HYPERLINK}* where *linkToSheet* is a text value in the following format: *".myLabelPropery||.myURLProperty" hyperLinkType* is one of the values: EMAIL, URL, SHEET or <empty>. For example: *".pyID|URL|.pyURLContent"*
Supported Pega Clipboard Data Pages	Text	Yes	Yes	
	Decimal	Yes	Yes	
	Double	Yes	Yes	
	Integer	Yes	Yes	

©2025 Pegasystems Inc

Feature	Name	Supported in Pega Platform 8.1	Supported in Pega Platform 8.2 and later	Description
	Date	Yes	Yes	
	TimeOfDay	Yes	Yes	
	DateTime	Yes	Yes	
	TrueFalse	Yes	Yes	
Properties from any Pega Clipboard page	Page	Yes	Yes	Pega Platform supports only single-level page lists, for example in the following format: *{.pxResuls().value input}* , *{.description input}*
	PageList	Yes	Yes	
Styles	Font, cell, color	Yes	Yes	Styling is not applied in all cases. If a single-cell style is defined, it will be applied, but styles for multiple cells are not retained. In the latter case, the application ignores the template styling and does not apply it to relevant cells or their content in the generated files.

©2025 Pegasystems Inc

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

Parameter	Supported in Pega Platform versions 8.1 to 8.3	Supported in Pega Platform 8.4 and later	Description
FSFileName	Yes	Yes	The name of the file to download.
TemplateRFB	Yes	Yes	Template details in the *Fileformat*`!`*Filename*`!xlsx` format.
DownloadFile	Yes	Yes	If the value is `true`, the application downloads the generated file.
DeleteFileAfterDownload	Yes	Yes	If the value is `true`, the server deletes the generated file after the application downloads the file.
evaluateFormulasLater	No	Yes	If the value is `true`, 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).

NOTE: If you have recently updated your Pega Platform instance that is hosted on a Weblogic server, running either the *pxParseExcelFile* or *pxGenerateExcelFile* activity might fail and cause the following exception in the logs:

©2025 Pegasystems Inc

```
java.lang.ExceptionInInitializerError: Exception java.lang.NoClassDefFoundEr
ror: Could not initialize class org.apache.poi.schemas.ooxml.system.ooxml.T
ypeSystemHolder [in thread "[ACTIVE] ExecuteThread: '4' for queue: 'weblog
ic.kernel.Default (self-tuning)'"]
```

This exception is caused by a dependency resolution issue on the Weblogic server side.

Related tasks

- Integrating a Case with an external resource

Related reference

- Case documents tags

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 Sliverlight. 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.

©2025 Pegasystems Inc

The following table provides information about support for Silverlight tags in the *pxGenerateWordDocument* API:

Tag in Silverlight	Support in the *pxGenerateWordDocument* API	Comments
{pega:reference propertyreference }	Yes	• Data references can vary as Silverlight reads data from BLOBs. • No support for localization in *pxGenerateWordDocument*
{pega:attachment attachfile="propertyName.insertMethod.extension.pzInsKey"}	No	• Use the tag *<pegaref config={"name":".AttachmentLink","type":"A"}>* instead. • *AttachmentLink* is the property reference that provides the *pzInsKey* of Data-WorkAttach-File or Rule-File-Binary. • No support for attachments other than images inside

©2025 Pegasystems Inc

Tag in Silverlight	Support in the *pxGenerateWordDocument* API	Comments
		tables, footers, and headers.
{pega:insert corr=corrname}	Partial	• No support for correspondence inside tables, headers and footers.
{pega:insert corrinline=corrname}	No	• Convert *{pega:insert corrinline=corrname}* to *{pega:insert corr=corrname}*. • Alternatively, to add correspondence inline within a document, use *<pegaref config={"name":"CorrespondenceName","type":"C"}>*. • No support for correspondence inside tables,

©2025 Pegasystems Inc

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

©2025 Pegasystems Inc

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.

Automatically moving a Case to a different Stage

Customize the order in which a Case proceeds. By revisiting or skipping a Stage, you can support out-of-sequence processing in a Case. For example, when a customer changes jobs during a loan request Case, the Case returns to a Stage in which the Case worker reviews the loan risk.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the Stage from which you want to move the Case, click **Add Step to Process** > **More** > **Automations**, and then select a transitioning Step for the Stage:

Choices	Actions
Automatically move the Case to a specific Stage	a. Click **Change to a Stage** > **Select**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **Step** properties pane, in the Stage list, select the Stage that you want the Case to enter.
Automatically move the Case to the next Stage	a. Click Change to Next Stage > Select.
Automatically move the Case to the previously visited Stage	a. Click Change to Previous Stage > Select. b. **Optional:** To enable returning to the Alternate Stage if it was the last visited Stage, in the **Step** properties pane, select the Consider previous Alternate Stage checkbox. **NOTE:** By default, the checkbox is cleared and the system moves a Case to the previously visited Primary Stage even if the previous Stage is an Alternate Stage.

NOTE: If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click Configure Process, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

©2025 Pegasystems Inc

3. **Optional:** To display a message with additional comments about the change in the audit trail of the Stage, in the **Step** properties pane, in the **Audit note** field, enter a Rule that stores the audit note that you want to include.
4. **Optional:** To provide a link to the current Stage in the breadcrumb trail navigation, in the **Step** properties pane, select the **Enable navigation link** checkbox, and then select any of the additional navigation options:
   - To enable users to get back to the Stage after the Case moves, select the **Only allow navigating back to this Step** checkbox.
   - To enable post-processing or validation when users navigate back from the Stage in the breadcrumb trail, select the **Perform post-processing when navigating away from Step** checkbox.
5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, the system resolves open Assignments before the Case leaves the current Stage. All Processes that originated from the Stage, including supporting Processes and excluding spin-off Processes, stop.

- Stages in a Case Lifecycle
- Steps in a Case Lifecycle
- Process shapes

PEGA ©2025 Pegasystems Inc

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. The main wait condition types that you can define are timer and Case dependency. 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 Type that describes the hiring process for a job candidate, and then create child Case Types that represent running a background check and collecting all of the required documents. You can then configure the parent Case Type to pause Case processing until one or all of the child Cases get resolved.

Before you begin:

For the Case dependency configuration in a Wait shape, add at least one Child Case Type to your parent Case Type before the Wait shape.

For more information, see Creating a child Case Type in Case Designer, Enabling automatic creation of child Cases, and Sharing data between parent and child Cases.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

CAUTION: If, at run time, Case processing reaches the Wait shape and no dependent Cases are created, Case processing continues from the Wait shape to the next Step. To change this default behavior, enable a configuration option for evaluating wait conditions even if no child

©2025 Pegasystems Inc

Cases have been created. You set the new default behavior in Process Modeler in the properties pane of the Wait shape by selecting the **Evaluate wait conditions, even if no child Cases are created** checkbox:

Wait shape configuration changing the new default behavior

For more information see Configuring complex Processes and Process shapes.

1. Add a Wait shape to the lifecycle of your Case Type:
   a. In the navigation pane of App Studio, click **Cases**, and then click the Case Type that you want to open.

©2025 Pegasystems Inc

b. In a Stage in which you want to add a Wait shape, click **Step** > **More** > **Automations** > **Wait**, and then click **Select**.

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

2. Define the conditions for resuming the Process:

Choices	Actions
Wait until a set amount of time passes	a. In the **Wait type** list, select **Timer**. b. Click **Set date/time interval**. c. Enter integers in one or more of the fields that are displayed, based on the interval of time that you want the Process to wait. **For example:** You can pause processing for a number of months, or you can be more specific by providing a number of hours and minutes.
Wait until a specific date or time	a. In the **Wait type** list, select **Timer**. b. Click **Reference date/time**. c. In the **Date/Time** field, define when you want to resume the processing.

©2025 Pegasystems Inc

Choices	Actions
	Ensure that you select a property that references the date and time in the future. For more information, see Expression Builder. **For example:** You can define the date by selecting a property from the list or clicking **Build an Expression** next to the **Date/Time** field, and then creating an Expression that returns a date and time value. Ensure that you select a property that references the date and time in the future. For more information, see Properties and Expression Builder.
Wait until all child Cases are resolved for every child Case Type referenced in the parent Case Type before the Wait shape	a. In the **Wait type** list, select **Case Dependency**. b. Select the **Wait for all child Cases to be resolved** checkbox.
Wait until one or all Cases of a specified child Case Type are resolved	a. In the **Wait type** list, select **Case Dependency**. b. In the first of the **Wait for (Case Type)** lists, select **Any** or **All** to indicate how many child Cases must

©2025 Pegasystems Inc

Choices	Actions
	be resolved before the Process resumes. c. In the second of the **Wait for (Case Type)** lists, select a child Case Type from which the child Cases are created. **TIP:** To select a Case Type that is not in the list, select **[Other]**, and then, in the **Class** field, provide a class that defines the Case Type. d. Click **To be resolved**. **NOTE:** A Case is considered to be resolved when its Case Status starts with the word `"Resolved-"`. e. **Optional:** To consider statuses only after the Case reaches the Wait shape, select the **Consider status after wait begins** checkbox. f. If the child Case Type has more than one parent in your Case Type-hierarchy, in the **Scope** list, select an

©2025 Pegasystems Inc

Choices	Actions
	option that identifies the instance on which the Process waits. **NOTE:** By default, the Wait shape pauses until the resolution of child Cases in the current Case Scope as well as in the descendant Case Types of the child Case, if any are configured. To exclude the descendants of the child Case Type from the wait condition, go to the exclude the descendants option. For more information, see Selective dependency check in the Wait shape configuration: use case. **TIP:** You can use the Case Type Explorer to review the structure of your Case Type hierarchy.
Wait until one or all Cases of a specified child Case Type reach a specific status	a. In the Wait type list, select Case Dependency.

©2025 Pegasystems Inc

Choices	Actions
	b. In the first of the **Wait for (Case Type)** lists, select **Any** or **All** to indicate how many child Cases must reach a specific status before the Process resumes. c. In the second of the **Wait for (Case Type)** lists, select a child Case Type from which the child Cases are created. **TIP:** To select a Case Type that is not in the list, select **[Other]**, and then, in the **Class** field, provide a class that defines the Case Type. d. Click **To reach status**, and then, in the field that is displayed, select a status value. e. **Optional:** To consider statuses only after the Case reaches the Wait shape, select the **Consider status after wait begins** checkbox. f. If the child Case Type has more than one parent in your Case Type hierarchy, in the **Scope** list, select an option that identifies the instance on which the Process waits.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: By default, the Wait shape pauses until the child Cases reach a specific status in the current Case Type Scope as well as in the descendant Case Types of the child Case, if there are any. To exclude the descendants of the child Case Type from the wait condition, go to the exclude descendants option. For more information, see Selective dependency check in the Wait shape configuration: use case. **TIP:** You can use the Case Type Explorer to review the structure of your Case Type hierarchy.
Wait until all Cases of a specified child Case Type are resolved within the Scope of only the current child Case Type	a. In the Wait type list, select Case Dependency. b. In the first of the Wait for (Case Type) lists, select All. c. In the second of the Wait for (Case Type) lists, select a child Case Type

©2025 Pegasystems Inc

Choices	Actions
NOTE: By default, the Wait shape pauses until the resolution of child Cases in the current child Case Type Scope as well as in the descendant Case Types of the child Case Type, if there are any. For more information, see Selective dependency check in the Wait shape configuration: use case.	from which the child Cases are created. **TIP:** To select a Case Type that is not in the list, select [Other], and then, in the Class field, provide a class that defines the Case Type. d. Click To be resolved. **NOTE:** A Case is considered to be resolved when its Case Status starts with the word `"Resolved-"`. e. **Optional:** To consider statuses only after the Case reaches the Wait shape, select the Consider status after wait begins checkbox. f. In the Scope list, select Current Case. g. Select the Evaluate wait dependency only on the current Case checkbox.

©2025 Pegasystems Inc

Choices	Actions
Wait until all Cases of a specified child Case Type reach a specific status only within the Scope of the current child Case Type **NOTE: By default, the Wait shape pauses until the child Cases reach a specific status in the current child Case Type Scope, as well as in the descendant Case Types of the child Case Type, if there are any. For more information, see Selective dependency check in the Wait shape configuration: use case.**	a. In the Wait type list, select Case Dependency. b. In the first of the Wait for (Case Type) lists, select All. c. In the second of the Wait for (Case Type) lists, select a child Case Type from which the child Cases are created. **TIP:** To select a Case Type that is not in the list, select [Other], and then, in the Class field, provide a class that defines the Case Type. d. Click To reach status. e. **Optional:** To consider statuses only after the Case reaches the Wait shape, select the Consider status after wait begins checkbox. f. In the Scope list, select Current Case. g. Select the Evaluate wait dependency only on the current Case checkbox.

3. **Optional:** To prevent Case workers from canceling the wait conditions and continuing to process a Case, uncheck the Users can choose to continue Process checkbox.

©2025 Pegasystems Inc

4. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a sample configuration of a Wait Step in a Case Lifecycle. The Step is configured with the Case dependency condition, which resumes parent case processing when at least one child Case of the Candidate interview type is resolved. The option for Case workers to skip the wait condition and continue the main Case processing is left selected:

Configuring a Wait shape to resume the primary Case processing after a specific child Case is resolved

©2025 Pegasystems Inc

What to do next:

Complement a paused Process with an optional Action to enable Case workers to handle related Assignments while waiting for the Process to continue. For more information, see Adding optional Actions to paused Processes.

- **Selective dependency check in the Wait shape configuration: use case**

- Adding optional Actions to paused Processes
- Case Lifecycle elements
- Assignment shapes in Processes
- Building Expressions with the Expression Builder

Selective dependency check in the Wait shape configuration: use case

The following scenario can help you learn how to use the selective dependency check in the Wait shape to avoid unnecessary dependency checks on descendant Cases, which improves workflow efficiency.

In onboarding processes, such as collecting due diligence information for a Contracting Party, you can use the selective dependency check in the Wait shape to improve efficiency. For example, in a process for collecting due diligence information when managing the onboarding of customers, also known as a Contracting Party (CP), the **CPKYC** Case Type is the parent and includes **eScreening** and **Adversemedia** child Case Types for both CPs and their Related Parties (RPs). One customer can have multiple RPs, for which you also need to run the **eScreening** and **Adversemedia** child Cases.

The following diagram shows the Case dependency in the due diligence collection business process:

©2025 Pegasystems Inc

The dependency structure of the due diligence collection Cases

If you do not want to halt the parent Case processing until the **RPGKYC** child Cases are completed, you can configure the Wait shape to wait only for the completion of the **eScreening** and **Adversemedia** child Cases in the current Case scope before moving to the next Stage, so that the primary KYC Case for each CP progresses to the Data Collection Stage without waiting for the resolution of screening Cases for the RPs.

By configuring the Wait shape to only check for eScreening and Adversemedia child Cases within the current CP Case, the parent Case progresses without waiting for related parties' screenings.

The following figure shows the configuration of the Wait shape to check for the **eScreening** child Case completion only in the current child Case context:

©2025 Pegasystems Inc

A Wait shape configured to resume after specific child Cases are started from the current Case, and not when its descendants are resolved

Related tasks

- Pausing and resuming Processes in Cases

Populating Case data with a Data Transform

Save time and speed up Case processing by prepopulating data in your Case. When you add the Run Data Transform Step to a Case Lifecycle, your application converts data from an external source to a format that fits your Case.

For example, while processing a purchase order, you can populate the shipping address with the data that the user provides as their billing address.

For more advanced use cases and training materials, see the Application data manipulation module and Manipulating application data challenge on Pega Academy.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

©2025 Pegasystems Inc

Before you begin:

- Create a Data Transform that you want the Case to call. For more information, see Data Transforms.
- Mark the Data Transform as a relevant record in your application. For more information, see Marking a record as relevant.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the Process in which you want to call a Data Transform, click **Step** > **More** > **Automations** > **Run Data Transform**, and then click **Select**.

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

3. In the **Step** properties pane, in the **Run Data Transform** list, select the Data Transform that you want to call in this Step.
4. **Optional:** In the **Audit note** field, enter some helpful extra text.

> **For example:**
>
> Enter `Data Transform updated`.

5. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

©2025 Pegasystems Inc

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Referencing properties
- Configuring a Data Transform
- Propagating data between parent and child Cases

Populating Case data with personalized AI-generated information

Leverage the power of generative AI Automation within your Pega Platform™ application to enhance user interactions and streamline workflow efficiency. When you add the **GenAI Connect** Step to a Case Lifecycle, you can swiftly populate data fields with personalized, AI-generated information, boosting productivity and delivering tailored responses to user prompts. For example, using **GenAI Connect** you can automatically replace personally identifiable information with placeholders in generative AI prompts, helping your organization enforce their data protection policies and advancing secure use of public and private models.

> **Before you begin:**
>
> Enable and configure generative AI in your application settings to use AI features. For more information, see Connecting to the generative AI provider in Pega Cloud.

1. Add the **GenAI Connect** Automation Step to the Case Lifecycle:
   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 Step, click **Step** > **More** > **Automations** > **GenAI Connect**.
   c. Click **Select**.

©2025 Pegasystems Inc

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

2. In the **Step** properties pane, configure the **GenAI Connect** Step:

Choices	Actions
Create a custom Generative AI connection	a. Select **Custom**. b. In the **Request** section, in the **Prompt** field, enter a prompt to instruct generative AI on the content to generate. c. Expand the **Response** section, and then, in the **Expected Response Mode** field, select the type of structure in which you expect to receive the response. > **NOTE:** For more information, see the response definitions in step 9 in Creating a GenAI Connect Rule. d. In the **To field(s)** field, select or create a field.
Use an existing Generative AI connection	a. Select **Use Existing**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **GenAI Connect name** field, enter the name of the existing *GenAI Connect* Rule that you want to call. **NOTE:** For more information about configuring AI Rules, see Creating a GenAI Connect Rule.

3. **Optional:** To provide additional comments about the connection, in the **Audit note** field, enter some relevant, helpful text.
4. **Optional:** To enable users to navigate to this Step from the Case breadcrumb navigation, select the **Enable navigation link** checkbox.
5. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, your application calls the generative AI Automation and displays the suggested data.

©2025 Pegasystems Inc

Posting messages to a Case

Inform Case workers about updates to a Case and provide a visible discussion thread for stakeholders by posting Pulse messages to a Case. By posting messages, you provide additional relevant information that enriches Case processing and helps you move your Cases faster towards resolution.

For example, when reviewing a job candidate, you can post a message that informs other involved stakeholders about the result of the recruitment process.

1. Add the **Post to Pulse** Step to the lifecycle of your Case:
   a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
   b. In the Process where you want to post Pulse messages, click **Step** > **More** > **Automations** > **Post to Pulse**.
   c. Click **Select**.

   > **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

   d. **Optional:** To provide a unique name for the Step, in the **Post to Pulse** field, enter a value.

   > **For example:**
   >
   > Enter `Inform about results`.

2. In the **Step** properties pane, identify the user who is responsible for the post:
   - To create a post for the user who currently processes the Case, in the **User posting** list, select **Current operator**.

©2025 Pegasystems Inc

- To create a post for another user, in the **User posting** list, select **Other**, and then select the ID of the user to post the message.

3. In the **Message** field, enter the text for the post.

> **For example:**
>
> Enter `The candidate has been approved for the position.`

4. **Optional:** To provide more information for auditing purposes, in the **Audit note** field, enter additional text.

> **For example:**
>
> Enter `The recruitment Process is completed.`

5. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **For example:**
>
> The following figure shows a message in the Pulse section for a Case related to reviewing a candidate:

©2025 Pegasystems Inc

Pulse *message in a Case*

- Case Lifecycle elements
- Collaborating with users by using Pulse

Preloading a Data Page

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

In your Case Lifecycle, add the **Load Data Page** Step before the Step that displays information to users. For example, add the **Load Data Page** Step that preloads currency rates before the **Calculate interest rate** Step in which a customer service representative (CSR) calculates rates for customers who want to obtain a loan in another currency.

1. Add the **Load Data Page** Step to a Case Lifecycle:
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. In a Process where you want to preload a Data Page, click **Step** > **More** > **Automations** > **Load Data Page**.

©2025 Pegasystems Inc

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

c. Click **Select**.

d. To provide a unique name for the Step, in the **Load Data page** field, enter a value.

> **For example:**
>
> Enter `Load currency rates`.

2. In the **Step** properties pane, in the **Data Page Name** field, press the Down arrow key, and then select the data page that you want to preload.
3. If the Data Page contains parameters, in the **Parameters** section, specify values for the parameters.
4. **Optional:** To provide more information for auditing purposes, in the **Audit note** field, press the Down arrow key, and then select an audit note to associate with the Step.
5. **Optional:** To enable users to navigate to this Step from the Case breadcrumb navigation, select the **Enable navigation link** checkbox.
6. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
7. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

For example:

The following figure shows a Step that preloads a Data Page with currency rates before a CSR calculates loan interest rates in a Case:

Preloading a Data Page Step

- Data Page definition
- Creating a Data Page
- Configuring load management options for data pages

Propagating data between parent and child Cases

Ensure data consistency between Cases by using an automated Step to propagate information automatically. For example, when a parent Case priority changes, configuring the **Update a Case** Step ensures that this updated priority is immediately applied to all associated child Cases, which streamlines workflows and reduces manual updates.

1. Add the **Update a Case** Step to the lifecycle of your Case:
    a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.

©2025 Pegasystems Inc

b. On the **Workflow** tab, in the **Lifecycle** view, in a Stage, click **+ Step**, and then click > **More** > **Automations** > **Update a Case** > **Select**.

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

Result:

The following figure shows the newly added **Update a Case** Step in a Case Lifecycle:

Update a Case Step in a Case Lifecycle

2. Click the newly added Step and then indicate which Cases to update:

Choice	Action
Update instances of a selected Case Type	a. In the Step property pane, select **A single Case**.

©2025 Pegasystems Inc

Choice	Action
	b. In the **With ID** field, reference the *InsKey* property from the work page of your Case Type. You can find work pages and properties related to your Cases in the Clipboard tool. For more information, see Clipboard tool. **For example:** To update a parent Case from a child Case, enter `.pxCoverInsKey`, which denotes the parent Case identifier within a child Case. c. In the **Data Transform** field, click the **Target** icon and configure a Data Transform that maps properties between the Cases. For more information, see Data Transforms. **For example:** The following figures show a sample Data Transform configuration for mapping properties from the child to the parent Case:

©2025 Pegasystems Inc

Choice	Action
	 Definition tab of a sample Data Transform with the mapping of child Case (source) properties to parent Case (target) properties. Specifying the child and parent work pages on the **Pages & Classes** tab of the Data Transform: *Child and parent work pages specified in the Data Transform*
Update instances of all the child Case Types and their descendants	a. In the header of your studio, click the **Switch Studio** menu, and then click App Studio. b. In the Step configuration pane, in the **Update child Cases** field, select **All** . c. Click the **Target** icon to select target fields and values to apply from the current Case.

©2025 Pegasystems Inc

Choice	Action
	For example: Use the operator and create time of a Case as the operator and update time of all its child Cases, as shown in the following figure: *Configuring the fields to associate with child Cases* d. Click **OK**.
Update instances of a specific child Case Type	a. In the header of your studio, click the **Switch Studio** menu, and then click App Studio. b. In the Step property pane, in the **Update child Cases** field, select a child Case from the list. c. Click the **Target** icon to select target fields and values to apply from the current Case. **For example:** Use the operator and create time of a Case as the

©2025 Pegasystems Inc

Choice	Action
	operator and update time of the child Case, as shown in the following figure: *Configuring the fields to associate with child Cases* d. Click **OK**.

3. **Optional:** To provide more information for auditing purposes, in the **Audit note** field select an instance of the *Rule-Message* Rule to associate with the Step.
4. **Optional:** To provide a link to this Step from the breadcrumb trail navigation, select the **Enable navigation link** checkbox.

> **NOTE:** This option is available only in Dev Studio.

5. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
6. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

- Referencing properties
- Understanding Case hierarchy
- Step types
- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools
- Sharing data between parent and child Cases

Running a queued background process in a Case

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

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

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

> **Before you begin:**
>
> To use the dedicated queue processor, you need to create the Rule first. For more information, see Creating a queue processor Rule.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. On the **Workflow** tab, click **Lifecycle**.
3. Hover over a Process in a Stage, and then click **Step** > **More** > **Automations** > **Run in background**.

©2025 Pegasystems Inc

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

4. Click **Select**.
5. In the **Step** properties pane, in the **Type of queue** list, choose a type of the queue processor that you want to run:
   - To use a default queue processor, select **Standard**, and then in the **Activity name** field, provide an activity that the Rule runs.
   - To use a custom queue processor rule, select **Dedicated**, and then in the **Queue processor** field, provide the queue processor that you want to run.

   For more information about when to use standard or dedicated queues, see Queue processor Rules:
6. In the **Lock using** list, select a type of lock that you want to apply:
   - To open a top-level page that your application currently processes, select **Primary page**.

     Your application uses a *pzInskey* to find and lock the page.
   - To open a page that you define manually, select **Key defined on property**, and then in the **Property name** field, enter a property name that corresponds with the page that you want to open and lock.
   - To open a page without a lock from the database, select **None**.
7. **Optional:** To resolve the activity in a different context than the context that your application uses to queue the item, in the **Alternate Access Group** field, enter the name of an Access Group that you want to use to resolve the activity.
8. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
9. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

©2025 Pegasystems Inc

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Queue processor Rules
- Replacing an agent with a queue processor Rule for delayed messages
- Replacing an agent with a queue processor Rule for real-time messages
- Changing a queue processor Rule state
- Tracing a queue processor

Running a questionnaire in a Case

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

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

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

PEGA ©2025 Pegasystems Inc

Before you begin:

- Create a questionnaire that you want to run in a Case. For more information, see Creating a questionnaire.
- If you update your questionnaires from Pega Platform 8.5 or earlier and you want to access questionnaire data from the parent Case, first remove and then reconfigure the **Questionnaire** Step in the parent Case, and next edit the *DataTableEditorReport* report definition by adding the *parentKey* parameter to a filter condition. Ensure that the *parentKey* parameter equals to the *.pxCoverInsKey* column source. For more information, see Report definitions.

1. Add the **Questionnaire** Step to the lifecycle of your Case.
   a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
   b. On the **Workflow** tab, in the **Case Life Cycle** section, hover over a Process, and then click **Step** > **More** > **Automations** > **Questionnaire**.

   NOTE: If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

   c. Click **Select**.
2. In the **Create Questionnaire** field, provide a name for the Step, for example, `Rate customer service`.
3. In the **Step** properties pane, in the **Select questionnaire** list, select the name of the questionnaire that you want to run as a child Case.

©2025 Pegasystems Inc

4. **Optional:** To reuse information from the parent Case, transfer data from the current Case to the questionnaire:

Choices	Actions
Map fields from the Case Type to the questionnaire	a. In the **Step** properties pane, select the Transfer info to questionnaire Case checkbox. b. In the **Transfer information** dialog box, in the 'From' field column, select the field with the values that you want to transfer. c. In the 'To' field column, press the Down arrow key, and then select a destination field to which you want to transfer the value. 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 checkbox. e. Click OK.
Configure advanced mapping by providing a Data Transform	a. Switch to Dev Studio. b. In the **Step** properties pane, select the Transfer info to questionnaire Case checkbox. c. In the list of transfer options, select Existing. d. In the Data Transform field, enter the name of the Data Transform that you want to apply.

©2025 Pegasystems Inc

Choices	Actions
	e. Click **Save**, and then switch to App Studio.

Result:

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

5. In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the questions in the questionnaire:
   - To route the questionnaire to the user who currently processes the Case, select **Current user**.
   - To route the questionnaire to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to complete the questionnaire.
   - To route the questionnaire to a Work Queue that multiple users share, select **Work Queue**, and then, in the **Work Queue** field, enter the name of the Work Queue.
6. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

After you add a Questionnaire Step to your parent Case Type, you can access the data that users provide only after the child Case for completing the

©2025 Pegasystems Inc

questionnaire is complete. In the Data Model of the parent Case, the system adds two new properties of a query type: a single property, and a list property. These two new properties source data from the questionnaire child Case Type. For more information about queries, see Querying a Data Page.

For example:

The following figure shows the effect of propagating data from a parent Case to a questionnaire child Case at run time in a process of reviving a reimbursement claim in an application for hotel operations. After a customer provides details, such as dates of arrival and departure, in the parent Case, when the questionnaire runs as a child Case, the application prepopulates answers with information that the customer already provided:

Prepopulating answers in a questionnaire

- Designing questionnaires

©2025 Pegasystems Inc

- Sharing data between parent and child Cases
- Fine-tuning your questionnaire

Sending event notifications from Cases

Notify users of your application about important Case events, for example, when a Case reaches a particular Stage in processing, by sending Case notifications.

By sending Case notifications, you can communicate with stakeholders in a format that is specific to their Channel, such as an email, a Portal, or a push notification. For more information, see Notification delivery Channels.

1. Add the **Send notification** Step to the Case Lifecycle:
   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 Lifecycle** section, in the Stage in which you want to add a **Send notification** Step, click **Step** > **More** > **Automations** > **Send notification**, and then click **Select**.

   > **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

2. In the **Step** properties pane, define the content and recipients of the notification:

Choices	Actions
Create a new Notification Rule	a. Select **Create new**. b. In the **Recipients** section, define how the system determines the list of recipients.

©2025 Pegasystems Inc

Choices	Actions
	For example: 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. **NOTE:** If you want to send the email based on the value of a field, you can select an embedded reference field with up to four levels of embedding, for example *.Customer.ProductionOrder.Recipient.EmailDetails.Email*. c. **Optional:** To add more recipients, click **Add recipient**, and then repeat step 2.b. d. In the **Message** section, in the **Message/Email subject** field, select the value that stores the push

©2025 Pegasystems Inc

Choices	Actions
	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. In the **Email body** field, select the custom or templated email option. **For example:** You can use a template for your notification message by selecting a Correspondence Rule or creating a new Correspondence Rule. For more information, see Creating a Correspondence Rule. f. If you selected **Custom message** in step 2.e, only for the email notification channel, in the **Email content** section, click **Compose**, and then compose the email message. For more information about composing email messages, see

©2025 Pegasystems Inc

Choices	Actions
	step 7 in Sending email notifications about Assignments routed to a user's Worklist.
Reuse an existing Notification Rule	a. Select **Use existing**. b. In the **Notification name** list, select the Notification Rule that you want to use.

3. **Optional:** To display a message in the audit trail of the Case, in the **Audit note** field, enter some helpful extra text or select a Rule that stores the audit note that you want to include.

   Result:

   Enter `Case updated with a Send notification Step..`

4. To enable users to navigate to this Step from the Case breadcrumb navigation, select the **Enable navigation link** checkbox.
5. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Steps in a Case Lifecycle
- Setting notification preferences
- Engaging and notifying stakeholders
- Creating a Notification Rule

PEGA ©2025 Pegasystems Inc

Sending automatic emails from Cases

By using email, you can share information about a Case with stakeholders and Case participants. Add the **Send Email** Step 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 candidate in a hiring process.

Before you begin:

- Ensure that you have an outbound email account configured in your system. For more information, see Configuring outbound email and Integrating your application with an email provider.
- When your application sends an email, the system updates the *pxSendDateTime* property on the *pxCorrSummary* work page that belongs to *pyWorkPage*. Such updates might cause issues with refreshing a Case at run time. To prevent errors, the *pyEnableCorrSummary* When Rule is set to `false` by default, and the system does not update the *pxSendDateTime* property. If your business scenario requires updating the *pxSendDateTime* property, in Dev Studio set the *pyEnableCorrSummary* When Rule to `true`.

1. Add the **Send Email** Step to the lifecycle 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 Lifecycle** section, in the Stage in which you want to add a **Send email** Step, click **Step** > **More** > **Automations** > **Send email**, and then click **Select**.

   NOTE: If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting

©2025 Pegasystems Inc

> Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

2. In the **Step** properties pane, identify the email recipients:
   - To send the email to one recipient, in the **Send to** list, select **Email address**, and then enter the email address of a user who wants to receive information about your Case.
   - To send an email to more than one recipient, enter the email addresses and separate them with commas.
   - To send the email based on the value of a field, in the **Send to** list, select **Field**, and then select the name of a field that stores an email address.

     > **NOTE:** You can select an embedded reference field with up to four levels of embedding, for example *.Customer.ProductionOrder.Recipient.EmailDetails.Email*.

   - 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 an 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** checkbox.

   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.

©2025 Pegasystems Inc

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, 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. 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.

©2025 Pegasystems Inc

Choices	Actions
	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. 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** checkbox. d. Enter the display text and title for the link. e. Click **OK**.
Generate the message with Autopilot	a. In the **Compose message** window, click **Generate message with Autopilot**. b. Verify the AI-Generated text for accuracy and modify content as needed.

©2025 Pegasystems Inc

Choices	Actions
	c. **Optional:** To regenerate the AI-generated message, click the **Menu** icon, and then click **Regenerate**. d. **Optional:** To regenerate the AI generated message with more context, click the **Menu** icon, and then click **Regenerate with more context**. In the **Provide additional context** field, enter additional information about your desired message content, then click **Regenerate**. **NOTE:** You must enable and configure Pega GenAI in your application settings to use these features. For more information, see Connecting to the generative AI provider in Pega Cloud.

7. Click **Done**.
8. **Optional:** To add attachments to the email, select the **Include attachments** checkbox, 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**.

©2025 Pegasystems Inc

Actions	Choices
	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.

9. In the upper-right corner of the Case Type, click **Save**.
10. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, when a Case reaches the **Send Email** Step, the application sends an email that you create to the recipients that you select.

For example:

The following video shows how to send an automatic email from a sample Case Type of reviewing a job candidate:

What to do next:

To speed up Case processing, configure email notifications about events. For more information, see Notifying Case participants about Case events.

- Adding an automated Step to a Process

©2025 Pegasystems Inc

- Categorizing Case attachments
- Enabling creation of Cases by email
- 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.

When you configure push notifications, the system ensures that the notification reaches the intended recipients through a two-step process:

1. Recipient identification

Based on the recipient type that you select, the system identifies which operators should receive the notification.

- **Application**: All users currently associated with the application.
- **Role**: Users in a specific Access Group.
- **User**: A specific user based on their identifier.

The system then determines the mobile devices associated with the identified operators. These devices are the ones the operators have registered by logging in to the application.

2. Notification delivery

Pega Platform™ sends notifications to the registered mobile devices using external push notification services:

- Apple Push Notification Service (APNs) for iOS devices.
- Firebase Cloud Messaging (FCM) for Android devices.

These services, provided by Apple and Google, ensure that notifications are delivered efficiently to the intended devices.

PEGA ©2025 Pegasystems Inc

1. Add the **Push notification** Step to the lifecycle 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** Step, click **Step** > **More** > **Automations** > **Push notification**, and then click **Select**.

   > **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

   c. Click **Select**.
2. In the **Step** properties pane, 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**.

©2025 Pegasystems Inc

Choices	Actions
	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** checkbox.
5. **Optional:** To enable users on mobile devices to open the Case directly from the notification, select the **Open a Case in mobile app** checkbox.
6. Enter an audit note.
7. In the upper-right corner of the Case Type, click **Save**.
8. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Case Lifecycle elements
- Steps in a Case Lifecycle

©2025 Pegasystems Inc

- Engaging mobile users with push notifications

Saving data in a Data Page as part of a Case Lifecycle

Manage data in your application more efficiently by saving Case data to a Data Page. As a result, you can save the data in a separate object, and then manage the data independently of the Case. Consequently, you speed up your application development and promote reuse of resources.

For example, in a financial application, if you save customer banking history as a Data Page, you can reuse the data in other Rules or pages in your application that require the use of a Data Page.

The source of information that you want to reuse needs to be a savable Data Page or an autopopulated property that points to a savable Data Page. After updating information in a Case, an application saves the information back to a selected Data Page.

> **Before you begin:**
>
> Configure a savable Data Page. For more information, see Creating a save plan for a Data Page in Dev Studio and Data Pages overview.

When you save information to a Data Page, you can either call the Data Page directly or use an autopopulated property. Autopopulated properties prevent your application from monitoring changes in key parameters as they automatically retrieve the object as soon as any property value of the object is needed and a key to the object is available. Completing these actions improves the performance of your application.

1. Add the **Save Data Page** Step to your Case Lifecycle:
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. In a Process where you want to preload a Data Page, click **Step** > **More** > **Automations** > **Save Data Page**.

PEGA ©2025 Pegasystems Inc

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

c. Click **Select**.

d. **Optional:** To provide a unique name for the Step, in the **Save Data Page** field, enter a value.

> **For example:**
>
> Enter `Save banking history`.

2. Configure how your application saves information to the Data Page:

Choices	Actions
Save data directly to the Data Page	a. In the **Step** properties pane, in the **Data Page Name** field, enter the name of the savable Data Page. b. **Optional:** If the Data Page has parameters, in the **Parameters** section, specify values for the parameters.
Use an autopopulated property	a. In the **Step** properties pane, select the **Use associated property** checkbox. b. In the **Autopopulate Property Name** field, enter the name of the property that you want to use.

©2025 Pegasystems Inc

3. **Optional:** To provide more information for auditing purposes, in the **Audit note** field, press the Down arrow key, and then select an audit note to associate with the Step.
4. **Optional:** To enable users to navigate to this Step from the Case bread crumb navigation, select the **Enable navigation link** checkbox.
5. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
6. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Related concepts

- Savable data pages
- Data Page definition
- Data pages and parameters

Spinning-off independent Cases from the current Case

Use a spin-off Case pattern when you want to automatically start an entirely independent Case from within the context of another Case, without creating a parent-child relationship.

This configuration is useful when the new Case needs to run on its own timeline, without depending on the originating Case's status or lifecycle.

For example, a fraud detection Case might trigger a separate investigation Case that proceeds independently but remains linked for visibility and traceability.

PEGA ©2025 Pegasystems Inc

> **Before you begin:**
>
> Create the main and related Case Types in your Constellation application. As a best practice, create the workflows with Pega GenAI Blueprint™. For more information, see Creating a new application from a Blueprint and Pega GenAI Blueprint - doc journey.

The goal of this sample configuration is to demonstrate a scenario in which the system automatically spins off an **Investigation** Case at the end of each **Alert** Case workflow.

Configuring a spin-off Case creation is a process that includes the following stages:

- Defining fields that store the relationship between the main and spun-off Cases
- Configuring automatic start of an Investigation Case from an Alert Case
- Ensuring that the new Investigation Case captures the Alert Case ID correctly by configuring a Data Transform
- Ensuring that the Alert Case properly stores and retrieves the Investigation Case ID by configuring a Data Transform

Defining fields that store the relationship between the main and spun-off Cases

1. In the Data Models of the Case Types, add and configure two Case Reference fields:
   The following steps show how to configure Reference fields to establish a relationship between the main and spun-off Cases. The steps cover both the Query and Case Reference types of fields. The Case Reference field is the most appropriate property type to use. However, when there is a back-and-forth relationship between two Cases, using Case Reference fields in both Cases might introduce infinite looping when loading one of the Cases. As a best practice, configure the reference as a Case Reference field in one Case and as a Query in the other.
   a. In the navigation pane of App Studio, click **Case Types**, and then select the main Case Type.

©2025 Pegasystems Inc

For example:

Click **Alert**.

b. On the **Data Model** tab, click **Add field**.
c. In the field configuration dialog box, in the **Field name** field, enter a unique label for the field.

For example:

Enter `AssociatedInvestigation`.

d. In the **Type** field, select **Case Reference**.
e. In the **Case Type** field, select the spun-off Case Type.

For example:

Select **Investigation**.

f. In the **Options** section, select **Single record**.
g. Click **Submit**.
h. In the upper-right corner, click **Save** to apply the changes.

Result:

The following figure shows the **Alert** Case Type Data Model with the field for referencing the **Investigation** Case added:

©2025 Pegasystems Inc

Field for referencing a linked Case from the main Case

i. In the navigation pane of App Studio, click **Case Types**, and then select the spun-off Case Type.

For example:

Click **Investigation**.

j. On the **Data Model** tab, click **Add field**, and then, in the field configuration dialog box, in the **Field name** field, enter a unique label for the field.

For example:

Enter `AssociatedAlerts`.

k. Enter the following values, and then click **Submit** > **Save**:
   - Type: **Query**.
   - Options: **List of records**.
   - Data Page: Data Page that you want to use in your Query field to fetch a list and the *pyIDs* of Cases of the main Case Type.

Result:

The following figure shows the **Investigation** Case Type Data Model with the field for referencing the **Alert** Cases added:

©2025 Pegasystems Inc

Field for referencing the main Case from the spun-off Case

For more information, see Referencing a Case Type and Querying a Data Page.

Configuring automatic start of an Investigation Case from an Alert Case

2. In the main Case, add a Step for starting the independent Case:
   a. In the navigation pane of Dev Studio, click **Case Types**, and then click the main Case Type.

> **NOTE:** Spinning-off independent Cases from the main Case Type is a Dev Studio exclusive feature.

> **For example:**
>
> Click **Alert**.

   b. On the **Workflow** tab, in the Process from which you want to start an independent Case, click **Add Step to the Process** > **More** > **Automations** > **Create Case**, and then click **Select**.
   c. In the **Create Case** Step field, enter the name of the Step.

> **For example:**
>
> Enter `Create Investigation`.

©2025 Pegasystems Inc

d. In the Step property pane, in the **Create** section, select **Case**, and then in the **Case Type** field, select the name of the Case Type that you want to start.

> **For example:**
>
> Select **Investigation**.

e. In the Create Case Step properties pane, in the **Property to store ID of Case** field, select `pyTempvalue`.

> **Result:**
>
> The following figure shows the **Alert** Case Type workflow with the Step for creating an **Investigation** Case added:

Step for creating an independent Case from the main Case workflow

Ensuring that the new Investigation Case captures the Alert Case ID correctly by configuring a Data Transform

3. In the main Case, in the **Create Case** Step, create a Data Transform to handle the Case ID transfer from the main to the spun-off Case:
   a. In the Step properties pane, in the **Data Transform** field, click the **Target** icon.
   b. Create a Data Transform Rule by performing steps 2 through 7 in Configuring a Data Transform.

©2025 Pegasystems Inc

> **For example:**
>
> Create the *UpdateAssociatedAlertsList* Data Transform.

c. On the **Definition** tab, in the first step, select the **Append and Map to** action, and then pass the following configurations:

Action	Target	Relation	Source
Append and Map to	`.`*<Query field from the spun-off Case>*	a new page	leave blank

This step is run after the primary Case is created. The step performs the following actions:

- The Query field in the spun-off Case, for example *.AssociatedAlerts*, is a list of references in the referenced, **Investigation**, Case Type.
- The step adds a new item to that list.

d. In substep 1.1 of the **Append and Map to** step, select the **Set** action and configure it with the following values:

Action	Target	Relation	Source
Set	*.pyID*	equal to	`pyWorkPage.pyID`

You configure this step in the Data Transform to populate the Case Reference property in the main Case with the ID of the spun-off Case.

Substep 1.1 sets *.pyID* to the value of *pyWorkPage.pyID*, which contains the ID of the main **Alert** Case.

You use *pyWorkPage* because:

- This step runs in the spun-off Case context, so the Primary Page is the spun-off Case.

©2025 Pegasystems Inc

- *pyWorkPage* holds the main Case data (source) in this background processing pattern.

e. Click **Save**.

Result:

The following figure shows the *UpdateAssociatedAlertsList* Data Transform with the **Append and Map to** step and **Set** substep configuration:

Data Transform steps for Case ID retrieval from the spun-off Case and population in the main Case

This Data Transform runs after the **Investigation** Case is created and is used to update the **Alert** Case with the ID of the **Investigation** Case.

f. Click **Save**.

Ensuring that the Alert Case properly stores and retrieves the Investigation Case ID by configuring a Data Transform

4. Add and configure the **Run Data Transform** automated Step to handle Case ID transfer from the spun-off Case to the main Case:
   a. On the **Workflow** tab of the main Case, after the **Create Case** Step that you added in step 2, click **Add Step to the Process** > **More** > **Automations** > **Run Data Transform**, and then click **Select**.
   b. In the Step properties pane, in the **Run Data Transform** field, click **Define a new Data Transform**.

©2025 Pegasystems Inc

c. In the **Data Transform name** field, provide a name, and then click **Create**.

> **For example:**
>
> Create the *SetCreatedInvestigationCaseRef* Data Transform.

d. Next to the **Run Data Transform** field, click the **Target** icon to open the Data Transform Rule form.

e. On the **Definition** tab of the Data Transform, in the first step, select the **Set** action, and then pass the following configurations:

This Data Transform runs after the spun-off Case is created and is used to populate the Case Reference field in the spun-off Case with the main Case ID.

Action	Target	Relation	Source
Set	*.<Case Reference field from the main Case>*`.pyID`	equal to	`@whatComesAfterLast(.pyTempValue, ' ')`

You configure this step in the Data Transform to populate the Case Reference property in the spun-off Case with the ID (*pyID*) of the main Case. Because *.pzInsKey* is returned by default when the Case is created, and the Case Reference expects *.pyID*, this data must be trimmed and mapped correctly.

This step is run after the primary Case is created, and then its *.pzInsKey* value is stored in the *.pyTempValue* temporary property. The step performs the following actions:

- Extracts the *.pyID* part.
- Sets that value in the Case Reference property, for example in *.AssociatedInvestigation*.

©2025 Pegasystems Inc

For more information, see Data Transforms.

Result:

The following figure shows the *SetCreatedInvestigationCaseRef* Data Transform with the Set step configuration:

Data Transform Set step for Case ID retrieval from the main Case and population in the spun-off Case

5. In the upper-right corner of the Case Type, click **Save**.
6. **Optional:** To test how your workflow behaves at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Referencing a Case Type
- Data relationships
- Configuring fields associated with Case and data objects
- Configuring a Data Transform
- Case Types
- Understanding Case hierarchy
- Building Case Types
- Enabling automatic creation of child Cases
- Sharing data between parent and child Cases

©2025 Pegasystems Inc

- Creating contextual Cases in traditional UI applications
- Starting child and Remote Cases from the main Case context

Extending a Data Model for a Case

Enhance the scope of data that your Case collects by adding fields to the Data Model. For example, to include a user email address in a Case, add an email address field to the Data Model.

Before you begin:

- Create a Case Type. As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

  For more information, see Creating a new application from a Blueprint.

- If you want to create a Data Model for a questionnaire, create the questionnaire first. For more information, see Creating a questionnaire.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Data Model** tab, click **Add field**.

   NOTE: You can also add new scalar fields to the Data Model from a View on the **User Interface** tab.

3. In the field configuration dialog box, in the **Field name** field, enter a descriptive name.

©2025 Pegasystems Inc

For example:

For example, enter `Email address`.

4. In the **Type** list, select a field type.
   For more information, see Configuring fields in the UI.
5. If more configuration options are available for the field type that you select, 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 **Display as** list, define whether to display the paragraph as plain or rich text. **Result:** The following figure shows a rich text paragraph at run time that users can use to describe symptoms while booking a doctor's appointment in an application: *Rich text paragraph*
Add a picklist to the form	a. In the **Type** list, select **picklist**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **Display as** list, define a display mode for the picklist. c. In the **Picklist options** list, define choices for the picklist. You can create your own choices by defining a local list or source items from a Data Page. In traditional UI applications, you can additionally source data from a Field Value. For more information see Creating Field Values. d. If you add your own choices, click **Add choice**, and then provide an option for users to select. **NOTE:** Configure your Case to enable Generative AI to select AI-suggested choices. For more information, see Connecting to the generative AI provider in Pega Cloud.

©2025 Pegasystems Inc

Choices	Actions
	 AI-generated picklist choices e. If you use a Data Page to provide choices, select the Data Page that you want to use. **Result:** The following figure shows a picklist with radio buttons, at run time, that users can use to select an office location: *Picklist*

©2025 Pegasystems Inc

Choices	Actions
	f. **Optional:** To predict the value of this field by using AI, configure additional settings in the **Advanced** section. This feature is available for Pega Process AI applications. See Classifying Cases by predicting field values.
Add an attachment field to the form	a. In the **Field name** field, type in a custom label for your field. b. In the **Type** list, select **Attachment**. c. In the **Attachment category** list, select the category or type in a custom label for your new category. d. In the **Type** list below the category, select **Single record** or **List of records**. **Result:** The following figure shows an attachment field at run time with which users can add documents to a Case: *Attachment field*
Add a user reference to the form	a. In the **Type** list, select **User reference**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **Display as** list, define whether users can search for a user ID by using a search box or a drop-down list. **Result:** The following figure shows a user reference field at run time, configured as a search box, that users can use to select a doctor while booking an appointment in an application: *User reference field*
Add a field to capture data	For more information about adding a field to capture data, see Creating fields for capturing data.
Add a field that references a Case Type	For more information about adding a field that references a Case Type, see Referencing a Case Type.

6. **Optional:** To add a help icon to your field, expand the **Advanced** note, and then, in the **Additional info** field, enter the content that you want to display in the help overlay.
   The **Additional info** field supports rich text, links, images, and field insertions.

©2025 Pegasystems Inc

Location of icons for adding fields, links, and images

7. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
8. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **What to do next:**
>
> Configure a View for your Case Type that includes the fields from the Case Type Data Model. For more information, see Designing the UI in Case Types.

- **Creating fields for capturing data**
- **Primary Fields**
- **Embedding data in a Case**
- **Referencing a Case Type**
- **Referencing a data object**

PEGA ©2025 Pegasystems Inc

- **Querying a Data Page**
- **Classifying Cases by predicting field values**
- Configuring Primary Fields
- UI components in a View
- Creating a Microjourney for customer success

Creating fields for capturing data

To decrease application development time and costs, reuse data objects across your application by creating fields that capture data. You can reference data between fields in several ways that help you to efficiently reuse resources without detailed knowledge about how the system stores data.

For example, in an **Account** data object that stores information that is related to customer accounts, you can create a **Contact** field to store contact information of a customer. To promote reuse, you can create a Case Type that uses the **Contact** field as the source of customer contact information.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

1. Navigate to a place where you want to add a field:

Choices	Actions
Add a field to the Data Model of a Case Type	a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open. b. On the **Data Model** tab, click **Add field**.

©2025 Pegasystems Inc

Choices	Actions
Add a field to a form in a View of a non-Constellation application	a. In the navigation pane of App Studio, click Case Types, and then click the Case Type that you want to open. b. On the Workflow tab, in the **Case Life Cycle** section, click an Assignment or an approval Step. c. In the **Step** properties pane, click Configure View. d. In the dialog box, on the Fields tab, click Add field.
Add a field to the visual Data Model of an application	a. In the navigation pane of App Studio, click Data. b. In the **Data Model** section, click View. c. In the Data Model, click a data object for which you want to create a data field. d. In the list of fields, click Create new field.
Add a field to a data object	a. In the navigation pane of App Studio, click Data. b. In the **Data objects** column, click the data object that you want to open. c. On the Data Model tab, click Add field.

2. In the field configuration dialog box, for the Field name parameter, enter a unique label for the field.
3. In the Type list, select Data Reference.

   The field captures single or multiple records that the system sources from a selected data object. The UI displays the field as a drop-down in which users can choose one (single record) or multiple (list of records) option from a given data object. For more information, see Configuring fields associated with Case and data objects.

4. In the Data object field, select or define the data object that you want to use as the source.
5. In the Options list, define how you want to reuse fields from the source:
   - To create one entry for all the fields from the source, select Single record.

©2025 Pegasystems Inc

- To create a list of separate entries for the fields from the source, select **List of records**.
- In a Constellation application, to create an object that models a many-to-many relationship, select **Many to many**.

6. **Optional:** To provide additional information about your data field, expand and complete the **Advanced** section:
   - To provide a different record ID than the autogenerated ID, in the **ID** field, enter a new value.
   - To provide additional information about the field, in the **Description** and the **Additional info** fields, enter more details.
7. In a Constellation application, if you select **Single record** or **List of records** in step 5, in the **Advanced** section, decide whether you can update the referenced data:
   - To prevent the system from copying any data from the Case to the system of record, leave the **Reference data is copied to the Case** checkbox clear.
   - To update data in the referenced object, select the **Reference data is copied to the Case** checkbox, and then in the **Data Page** field, select a savable Data Page that has the option to copy data from a Data Page enabled. For more information, see Creating a Data Page, Using Data Pages in a Case Lifecycle, Updating data in a Case by using savable Data References, and Using data pages in forms.
8. Save your data field:
   - To save the field and instantly define another field, click **Submit & add another**.
   - To save your field and return to the configuration view, click **Submit**.
9. Save your changes by clicking **Save** or **Save and run**.

What to do next:

Define how the field that you add to a form behaves inside the View. For more information, see Configuring a single-record reference field and Configuring a list of records reference field.

©2025 Pegasystems Inc

- Creating Views for Case Types
- Populating Views

Primary Fields

Primary Fields represent the critical pieces of data that you want to see whenever you interact with a Case of a specific type. With Primary Fields, low-code application developers can quickly configure a Case Type with an automatically-generated set of default Views.

> ⚠ **IMPORTANT:** Only applications that you build with Constellation support Primary Fields. For more information, see Application UI setup.

The aim of Primary Fields is to ensure a consistent user experience for end users when they interact with a Case, whether they are creating a new Case, editing, or viewing a summary of an existing Case. To achieve this, in the Data Model of Case Type, you mark the fields that are the most relevant to a Case Type as Primary Fields.

For example, application developers may use Primary Fields to define key fields for a vehicle and its owner in the Vehicle Registration Case Type. In this Case Type, the *Vehicle* field is marked as a Primary Field in the Case Type, and the following individual fields are marked as primary in the *Vehicle* Data Type: *Registration ID*, *Registration Date*, *Owner*, *Vehicle ID*, *Insurance Provider*, *Insurance Start Date*, *Insurance End Date*.

After you define the Primary Fields for your Case Type, all the following out-of-the-box Views by default display these fields, which reduces the need for additional UI configuration:

Create and Edit Views

Primary Fields are automatically available for your Case Type during the creation and editing of Cases that are instances of that Case Type.

©2025 Pegasystems Inc

Details View

The Details View in a Case automatically includes the Primary Fields for the Case Type.

List View

The List Views in your Case default to Primary Fields automatically.

The Data Model of a Case Type automatically designates the `Label` and `Description` fields as primary by default.

You can fine tune the Views later in the application development lifecycle. For more information, see Working with Views.

- **Configuring Primary Fields**

Related concepts

- Basic concepts in Data Modeling

Related tasks

- Configuring details Views
- Configuring list Views

Related reference

- Configuring fields in the UI
- Constellation taxonomy

Configuring Primary Fields

Define Primary Fields to automatically add the most relevant fields to your Cases and reduce the amount of UI configuration that is necessary.

PEGA ©2025 Pegasystems Inc

> **NOTE:** Only applications that you build with Constellation support Primary Fields. For more information, see Application UI setup.

You can add Primary Fields to Case Types and data objects without the need to configure the UI. Instead, you only modify your Data Model and the system automatically adjusts the form. Pega Platform™ automatically generates a **Primary Fields** View and adds it to the **Create**, **Edit**, **Details**, and **List** Views. Case Types include the **Label** and **Description** Primary Fields by default.

For example, in a Case Type that captures customer information, you define the **Name**, **Email**, and **Address** fields as Primary Fields. Then, you access the **Address** data object and define its Primary Fields, such as **Street**, **City**, and **Country**. The system then automatically adds the fields in the **Primary Fields** View, as shown in the following figure:

Sample Primary Fields preview

For more information, see Working with Views.

©2025 Pegasystems Inc

1. In the navigation pane of App Studio, navigate to the item for which you want to define Primary Fields:

Choices	Actions
Case Types	Click **Case Types**, and then click the Case Type that you want to open.
Data objects	Click **Data**, and then click the data object that you want to open.

2. On the **Data Model** tab, click **Primary Fields**.
3. In the **Primary Field configuration** dialog box, click **Add**, and then select the existing fields that you want to make primary.
4. **Optional:** To delete existing Primary Fields, click the **Remove from Primary Fields** icon.
5. **Optional:** To reorder the Primary Fields, drag and drop the fields into the appropriate positions.
6. Click **Submit**.
7. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
8. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Related concepts

- Adding fields

Related reference

©2025 Pegasystems Inc

- Constellation taxonomy

Embedding data in a Case

Manage your resources more efficiently by embedding data and connecting information directly to a Case. Consequently, you avoid sourcing data from any other object. Embedding relevant data directly in a Case if the Case is the only object that uses this piece of information helps you to organize data in your application in a more transparent way.

For example, you can create an *Address* field and embed that field in a Case Type if this Case Type is the only element in your application that uses the information from the *Address* field.

To embed data in a Case, you create a data field that stores the relevant information. You can add a data field to the following interfaces:

- The Data Model of a Case Type. For more information, see Extending a Data Model for a Case.
- A form in a View for a Step in a Case Type for applications that do not use Constellation. For more information, see Adding single-value fields to forms.
- The visual Data Model of your application . For more information, see Exploring the visual Data Model.
- The Data Model of a data object. For more information, see Exploring the visual Data Model and Data objects overview.

1. Navigate to a place where you want to add a field:

Choices	Actions
Add a field to the Data Model of a Case Type	a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open. b. On the **Data Model** tab, click **Add field**.
Add a field to a form in a View	a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.

©2025 Pegasystems Inc

Choices	Actions
of a non-Constellation application	b. On the Workflow tab, in the **Case Life Cycle** section, click an Assignment or an approval Step. c. In the **Step** properties pane, click Configure View. d. In the dialog box, on the Fields tab, click Add field.
Add a field to the visual Data Model of an application	a. In the navigation pane of App Studio, click Data. b. In the **Data Model** section, click View. c. In the Data Model, click a data object for which you want to create a data field. d. In the list of fields, click Create new field.
Add a field to a data object	a. In the navigation pane of App Studio, click Data. b. In the **Data objects** column, click the data object that you want to open. c. On the Data Model tab, click Add field.

2. In the field configuration dialog box, for the Field name parameter, enter a unique label for the field.
3. In the Type list, select Embedded data.
4. In the Data object list, define a data object to store data:

Choices	Actions
Create a new data object	a. In the list, select Create a new data object. b. In the Data object name field, enter a descriptive label for the data object. c. Click OK.
Reuse a data object	In the list, select a data object that stores the relevant data that you want to embed in the Case Type.

©2025 Pegasystems Inc

Choices	Actions
	The list contains all data objects from your application.

5. In the **Options** list, define how you want to store the fields from the data object:
   - To create one entry for all the fields from the data object, select **Single record**.
   - To create a list of separate entries for the fields from the data object, select **List of records**.
6. **Optional:** To provide additional information about your data field, expand and complete the **Advanced** section:
   - To provide a different ID than the autogenerated ID, in the **ID** field, enter a new value.
   - To provide additional information about the field, in the **Description** text box, enter additional details.
7. Save your data field:
   - To save the field and instantly define another field, click **Submit & add another**.
   - To save your field and return to the configuration view, click **Submit**.
8. In the upper-right corner of the Case Type, save your changes by clicking **Save** or **Save and run**.

- Creating fields for capturing data
- Constellation UI
- Data Modeling
- Data objects overview

Referencing a Case Type

Build transparent and automatic relationships between Case Types by creating Case references. When you reference a Case Type, you can pass data about one Case Type to another Case Type, and you can clearly view connections between Case Types and their

©2025 Pegasystems Inc

instances at run time. You also save time and take advantage of low-code tools because you can view relations between Case Types without creating complex report definitions.

For example, you can reference the **Manage account** Case Type from the **Offer opportunity** Case Type. As a result, you can quickly view offered opportunities that are related to a managed account. You can also create a UI to display the offered opportunities at run time from the View of a Case for managing an account.

> **NOTE:** Reverse fields are added to referred Case Types by default, creating a two-way relation between the target Case and the referring Case.

When you reference a Case Type, the system automatically creates a field of the query type in the Data Model of the referenced Case Type. For example, if you reference the **Manage account** Case Type in the **Offer opportunity** Case Type, the system automatically adds a query field in the Data Model of the **Manage account** Case Type. As a result, you can clearly analyze connections between Case Types.

To adjust the processing and display of Case references at run time, you can decide whether a Case should reference other Cases as a single object, or a list of objects. For example, you can reference only one run-time instance of a **Manage account** Case Type from an **Offer opportunity** Case Type. Then, an application displays only one Case ID. You can also connect multiple instances of the **Offer opportunity** Case Type to the same instance of the **Manage account** Case Type, and configure an application to display a table with multiple Case IDs.

> **NOTE:**
>
> - Case references work most efficiently in Constellation applications.
>
> For more information, see Application UI setup.

©2025 Pegasystems Inc

- Only Constellation applications support Remote Case reference.

1. Navigate to a place where you want to add a field:

Choices	Actions
Add a field to the Data Model of a Case Type	a. In the navigation pane of App Studio, click Case Types, and then click the Case Type that you want to open. b. On the Data Model tab, click Add field.
Add a field to a form in a View of a non-Constellation application	a. In the navigation pane of App Studio, click Case Types, and then click the Case Type that you want to open. b. On the Workflow tab, in the **Case Life Cycle** section, click an Assignment or an approval Step. c. In the **Step** properties pane, click Configure View. d. In the dialog box, on the Fields tab, click Add field.
Add a field to the visual Data Model of an application	a. In the navigation pane of App Studio, click Data. b. In the **Data Model** section, click View. c. In the Data Model, click a data object for which you want to create a data field. d. In the list of fields, click Create new field.
Add a field to a data object	a. In the navigation pane of App Studio, click Data. b. In the **Data objects** column, click the data object that you want to open. c. On the Data Model tab, click Add field.

2. In the field configuration dialog box, for the Field name parameter, enter a unique label for the field.
3. In the Type list, select Case reference.
   For more information, see Configuring fields in the UI.
4. In the Case Type list, select a Case Type that you want to reference.

©2025 Pegasystems Inc

In Constellation applications, the list also displays remote Case Types.

5. In the **Options** list, define how you want to retrieve and display data from the source Case Type:
   - To create one entry for data from the source, select **Single record**.

     For example, you can connect an Opportunity Case Type to one instance of the Manage account Case Type. At run time, an application displays only one field for the referenced data.

   - To create a list of separate entries for data from the source, select **List of records**.

     For example, you can connect a Manage account Case Type to multiple instances of an Opportunity Case Type. At run time, an application displays a grid table with sourced information.

   - To create a reference that models a many-to-many relationship, select **Many to many**.

     > **NOTE:** This option is unavailable for a remote Case reference.

   > **For example:**
   >
   > For example, you can connect a Manage account Case Type to multiple instances of an Opportunity Case Type. At run time, an application displays a grid table with sourced information.

6. **Optional:** To copy the referenced data into a Case at run time, expand the **Advanced** section, and then select the **Reference data is copied to the Case** checkbox.

©2025 Pegasystems Inc

Copying data creates a snapshot of the referenced data and preserves the data in the target unchanged if data in the source changes. If you do not want to persist any changes of data to the system of record, ensure the checkbox is cleared.

> **NOTE:** As a result, in a Constellation application, if you select a Case Type that references a savable Data Page in step 4 and select the **Reference data is copied to the Case** checkbox, in Case Lifecycle authoring, the **Data Relationship** drop-down list in the **Step** property pane includes only fields that are configured with the copy-enabled savable Data Pages.

7. **Optional:** To provide another Data Page to source data, expand the **Advanced** section, and then, in the **Data Page** field, enter the Data Page that you want to use.

> **For example:**
>
> You can provide a Data Page that filters Case Types by status to display only Cases with an Active status at run time. For more information, see Creating a Data Page and Editing data pages.

> **NOTE:** For a Remote Case Type reference, the list displays only Data Pages from the Remote Case Type that have the alternate key storage parameter enabled. For more information, see Adding alternate keys for external reference fields.

8. **Optional:** To change the unique identifier of a referenced Case Type, expand the **Parameters** section, and then configure how an application sources the new value:
   - To manually provide a new value, in the **pyID** list, select **Constant**, and then, in the text field, enter a new unique identifier for the Case Type.

©2025 Pegasystems Inc

- To source the value from a field in the current Case Type, in the **pyID** list, select **Field**, and then, in the list, select a field that stores the value that you want to use as the unique identifier.
- To source the value from a field in the referenced Case Type, in the **pyID** list, select **Field in target**, and then, in the list, select a field that stores the value that you want to use as the unique identifier.

You can reference a unique identifier, which is also a *pyID* parameter, in various elements in your application, such as reports.

9. Save your data field:
   - To save the field and instantly define another field, click **Submit & add another**.
   - To save your field and return to the configuration view, click **Submit**.
10. In the upper-right corner of the Case Type, save your changes by clicking **Save** or **Save and run**.

Result:

The system creates a field of a query type in the Data Model of the referenced Case Type, as shown in the following figure:

©2025 Pegasystems Inc

Case reference field configuration

For example:

The following figure shows an instance of a Manage account Case Type at run time with the Opportunities section that an application sources by using a Case reference with a list of records configurations. At run time, the system automatically populates the fields with values that are provided by an instance of the referenced Case Type.

©2025 Pegasystems Inc

Case Type reference at run time

What to do next:

- Add a field for selecting a Case that you reference to the View of your Case Type. For example, in an Offer opportunity Case Type, add a field to select an instance of a Manage account Case. For more information, see Adding single-value fields to forms.
- Add a field that displays referenced Case Types, and then configure the display as a table. For example, in the Manage account Case Type, add a field to display instances of the Offer opportunity Case Type. For more information, see Configuring a list of records Embedded Data field.

- Designing the UI in Case Types
- Defining default data sources
- Configuring a single-record reference field
- Configuring a list of records reference field

Referencing a data object

Use data sourced from outside of a Case by creating a field that references a data object. A Data reference can connect to data sourced either from locally stored records or from an external system of record.

©2025 Pegasystems Inc

Unlike a data object, a reference does not store data. Rather, it relates data between the elements of your application that do store data and the elements of your application that need that data.

For example, in an **Account** data object that stores information that is related to customer accounts, you can create a **Contact** field to store the contact information of a customer. To promote reuse, you can then create a Case Type that uses the **Contact** field as the source of customer contact information.

1. Navigate to a place where you want to add a field.
   For more information, see step 1 of Creating fields for capturing data.
2. In the field configuration dialog box, for the **Field name** parameter, enter a unique label for the field.
3. In the **Type** list, select **Data Reference**.

   The field captures single or multiple records that the system sources from a selected data object. The UI displays the field as a drop-down in which users can choose one (single record) or multiple (list of records) option from a given data object. For more information, see Configuring fields associated with Case and data objects.

4. In the **Data object** field, select or define the data object that you want to use as the source.
5. In the **Options** list, define how you want to reuse fields from the source:
   - To create one entry for all the fields from the source, select **Single record**.
   - To create a list of separate entries for the fields from the source, select **List of records**.
   - In a Constellation application, to create an object that models a many-to-many relationship, select **Many to many**.
6. **Optional:** To provide additional information about your data field, expand and complete the **Advanced** section:
   - To provide a different record ID than the autogenerated ID, in the **ID** field, enter a new value.

©2025 Pegasystems Inc

- To provide additional information about the field, in the **Description** and the **Additional info** fields, enter more details.

7. In a Constellation application, if you select **Single record** or **List of records** in step 5, in the **Advanced** section, decide whether you can update the referenced data:
   - To prevent the system from copying any data from the Case to the system of record, leave the **Reference data is copied to the Case** checkbox clear.
   - To update data in the referenced object, select the **Reference data is copied to the Case** checkbox, and then in the **Data Page** field, select a savable Data Page that has the option to copy data from a Data Page enabled. For more information, see Creating a Data Page, Using Data Pages in a Case Lifecycle, Updating data in a Case by using savable Data References, and Using data pages in forms.
8. Save your data field:
   - To save the field and instantly define another field, click **Submit & add another**.
   - To save your field and return to the configuration view, click **Submit**.
9. Save your changes by clicking **Save** or **Save and run**.

What to do next:

Define how the field that you add to a form behaves inside the View. For more information, see Configuring a single-record reference field and Configuring a list of records reference field.

- Creating Views for Case Types
- Populating Views
- Data relationship for data reference
- Creating Views for Case Types
- Populating Views

PEGA ©2025 Pegasystems Inc

Querying a Data Page

Provide data for your Case Types from integrations in your application by querying a Data Page. When you source data for your Case directly from a Data Page, you reuse your resources in a transparent and prescriptive way.

For example, you can source information from a Data Page that has integration with a third-party system to display weather forecast in your application.

To reuse data from a Data Page in your Case Type, you create a data field that queries a Data Page.

By creating a field that references a Data Page you can reuse information that a Data Page stores, in interfaces that typically do not allow referencing Data Pages, for example a Condition builder or a decision table. As a result, you promote reuse of information across your application, save application development resources, and provide more flexible software. When you reference a Data Page, you receive entire content of the Data Page. If a Data Page has parameters, you can specify the parameters to receive precise information that your business process requires. For example, when you source information about a weather forecast, you can use a location as a parameter so that your application displays the forecast for a selected city.

1. Navigate to a place where you want to add a field:

Choices	Actions
Add a field to the Data Model of a Case Type	a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open. b. On the **Data Model** tab, click **Add field**.
Add a field to a form in a View of a non-Constellation application	a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open. b. On the **Workflow** tab, in the **Case Life Cycle** section, click an Assignment or an approval Step. c. In the **Step** properties pane, click **Configure View**.

©2025 Pegasystems Inc

Choices	Actions
	d. In the dialog box, on the **Fields** tab, click **Add field**.
Add a field to the visual Data Model of an application	a. In the navigation pane of App Studio, click **Data**. b. In the **Data Model** section, click **View**. c. In the Data Model, click a data object for which you want to create a data field. d. In the list of fields, click **Create new field**.
Add a field to a data object	a. In the navigation pane of App Studio, click **Data**. b. In the **Data objects** column, click the data object that you want to open. c. On the **Data Model** tab, click **Add field**.

2. In the field configuration dialog box, for the **Field name** parameter, enter a unique name, for example `Weather forecast`.
3. In the **Type** list, select **Query**.
   For more information, see Configuring query fields and Configuring fields in the UI.
4. In the **Options** list, define how you want to reuse the fields from the source:
   - To create one entry for all the fields from the source, select **Single record**.
   - To create a list of separate entries for the fields from the source, select **List of records**.
5. In the **Data Page** list, select the Data Page that you want to use as the source.
6. If the source Data Page has parameters, in the **Parameters** section, provide information about the parameters of the source Data Page:
   - To provide a static value to search for specific information in the Data Page, in the **pyID** list, select **Constant**, and then in the text field, provide the value that you want to use for searching the Data Page.
   - To select a field in the Data Page to source information, in the **pyID** list, select **Field**, and then, in the list, select the field that you want to use to source data.

©2025 Pegasystems Inc

7. To provide additional information about your data field, expand and complete the **Advanced** section:
   - To provide a different ID than the autogenerated ID, in the **ID** field, enter a new value.
   - To provide additional information about the field, in the **Description** text box, enter additional details.
8. **Optional:** To copy the referenced data into a Case at run time, expand the **Advanced** section, and then select the **Reference data is copied to the Case** checkbox.
   Copying data creates a snapshot of the referenced data and preserves the data in the target unchanged if data in the source changes. If you do not want to persist any changes of data to the system of record, clear the checkbox.

   > **NOTE:** In a Constellation application, if you select a savable Data Page in step 5 and the **Reference data is copied to the Case** checkbox. in Case Lifecycle authoring, the **Data Relationship** drop-down list in the **Step** property pane includes only fields that are configured with the copy-enabled savable Data Pages.

9. Save your data field:
   - To save the field and instantly define another field, click **Submit & add another**.
   - To save your field and return to the configuration view, click **Submit**.
10. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
11. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

   > **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

What to do next:

Configure the newly-added field to display data in the UI. For more information, see Configuring query fields.

- Data Page definition
- Data Modeling
- Creating fields for capturing data
- Constellation UI
- Embedding data in a Case
- Defining default data sources
- Extending a Data Model for a Case
- Adding single-value fields to forms
- Exploring the visual Data Model
- Data objects overview

Classifying Cases by predicting field values

Improve business processes by predicting the category of a Case, such as a claim, IT incident, or support issue. You can use the predicted category for intelligent routing to assign each Case to an appropriate Work Queue or to auto-complete the Case category when a Case worker manually triages Cases.

With AI-driven Case classification, you can automatically classify Cases and route them to the right Work Queue, so that you do not need to reassign them.

Case classification predictions are based on natural language processing (NLP). The system uses a topic detection model to analyze text that is associated with the Case, typically the Case description, and predict the Case category.

PEGA ©2025 Pegasystems Inc

Before you begin:

- Ensure that decision nodes are present in your system. For more information, see Pega Process AI installation.
- Create a Case Type. For more information, see Case Types.
- In the Case Type Data Model, create a field of the **Text (line)** or **Picklist** type with a meaningful name, such as *Accident Category*. For more information, see Extending 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. On the **Data Model** tab, click the **Gear** icon to the right of the field that you created.
3. The **Configure field** window, expand the **Advanced** section, and then perform one of the following actions:
   - For a text field, in the **Input method** list, select **Predicted with AI (editable)**.
   - For a picklist field, select the **Use AI to predict the value of this field** checkbox.
4. In the **Input text for AI analysis** field, select a text field that contains the input text for analysis, for example, **Description**.
   The topic detection model will predict the most likely value for the field. For a picklist, the model will predict the most likely value from the possible choices.
5. In the **Minimum confidence to use predicted value** field, enter the minimum confidence at which the field will display the predicted value.

   The field will only take the predicted value if the confidence is high enough. By default, confidence is set to 80%.

©2025 Pegasystems Inc

For example:

The following figure shows an example of a picklist field configuration that uses AI settings to predict the accident category based on the claim Case description:

©2025 Pegasystems Inc

AI settings for a picklist field

©2025 Pegasystems Inc

The following figure shows an example of a single-line text field configuration that uses AI settings to predict the Case category based on the Case description:

AI settings for a text field

6. Click **Submit**.

©2025 Pegasystems Inc

Result:

The field is now a part of the Data Model for the Case. You can use the predicted values to enhance any Step of a Case that uses a Condition, such as Assignment, Change to a Stage, or Decision Steps.

For example, you can define business logic to assign the Case to different Work Queues based on the predicted Case category, such as `Property Damage`, `Bodily Injury`, or `Uninsured or Underinsured Motorist`, as shown in the following figure:

©2025 Pegasystems Inc

Example of a business logic for routing Cases based on the predicted Case category

For more information, see Configuring custom routing logic for Assignments.

When you create a predicted field, Pega Platform™ creates a Text Prediction that includes the topic detection model for predicting the field value by analyzing the input text, for example, the Case category or incident location

©2025 Pegasystems Inc

based on the Case description. The link to the Prediction is available in the **Options** column for the field, as shown in the following figure:

Link to the Text Prediction for a predicted field

You can increase the accuracy of predictions by training the topic detection model with Case-related historical data. See Analyzing messages with text predictions.

7. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
8. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

Designing the UI in Case Types

Ensure that your business process has a proper user interface to display and collect information so that users can fulfill requests and quickly and intuitively process Cases towards resolution.

Pega offers two architectures that can help you bring value and meaning to your business processes. For more information on how to check your UI setup, see Application UI setup.

Depending on the UI architecture that you use in your application, see the following topics on how you can design the UI in Case Types:

- For Constellation applications: Constellation UI and Creating Insights from UI authoring.
- For traditional Section-based UI applications: Traditional UI.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

Related reference

- Choosing a UI

Restricting user input in a field

Configure fields in your Data Model to define the way that they store or display information to help users provide data in a format that is required in your business scenario. As a result, you minimize the risk of providing erroneous data and improve quality of Case processing.

PEGA ©2025 Pegasystems Inc

For example, you can create a calculated field that automatically sums up values in a Case. Consequently, you increase a level of automation in your Case Type and accelerate processing of Cases. You can also implement validations to ensure that users enter data in a correct format.

Explore the following articles to learn more about solutions that you can use in your Case Types:

- **Validating input values on a form**
- **Validating field input in complex scenarios**
- **Adding a Validation Rule to a Flow Action**
- **Validating character patterns**
- Creating Views for Case Types

Validating input values on a form

Ensure that users enter valid data on a form before the Case moves to the next Step in the Lifecycle. By validating field values, you prevent processing errors and speed up the Case resolution process. If the data that the user enters on a form at run time meets the Condition in a Validation Rule, the application displays an error message.

For example, you can ensure that the customer provides their name and address before placing an order. If the customer leaves the fields empty, the application displays an error message.

> **Before you begin:**
>
> Create a Case Type, and then define a Case Lifecycle that has Stages and Processes with Assignments. Add fields that you want to validate to the forms in your Assignments. For more information, see Automating work by creating Case Types.

©2025 Pegasystems Inc

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 Lifecycle** section, select the Step to which you want to apply the Validation Condition.
3. In the **Step** properties pane, 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**:

Choices	Actions
Mark a field as required in a traditional Pega Platform application	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, select **Required**. c. Click **Submit**. When the user tries to submit the form without a value in a required field, the application displays an error message.
Mark a field as required in a Constellation application	For more information about defining field behavior in Constellation, see Configuring field behavior.

> **For example:**
>
> To prevent the user from submitting a form without entering their date of birth, set the birth date field as **Required**.

5. In the View configuration window, define a new Validation Condition:
   - In a traditional Pega Platform application, click the **Validations** tab, and then click **Add Condition**.
   - In a Constellation application, click the **Conditions** tab, and then click **Add Condition**.

©2025 Pegasystems Inc

6. In the **Message** field, enter the text of an error message that is displayed when the Validation fails.
7. Define the Validation Conditions:

Choices	Actions
Define a Validation Condition for a field	a. In the list of values, select **Fields**, and then 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 to the user input. The `Select values` option lists the fields and values that you can use in the Condition.
Define a Validation Condition for a field that corresponds to a Page or Page list property property	a. In the list of values, select **Fields**, and then, next to the name of the field that you want to validate, click the **Add** icon. b. In the comparator list, select the test that you want to perform on the field. **For example:** To use an existing Validation, select **call a Validation**. c. In the value field, enter or select a value to compare to the user input.

©2025 Pegasystems Inc

Choices	Actions
	The Select values option lists the values and Rules that you can use in the Validation.
Apply an existing When Condition as a Validation Condition	a. In the list of values, select **When Conditions**, and then select the name of the When Condition that you want to apply. b. In the comparator list, select the test that you want to perform on the When Condition.

For example:

To ensure that the birth date is not 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 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
The Condition passes when all properties meet the criteria	a. Click **Add a row**. b. In the operator list, select **and**. c. In the list of fields, select the name of the field that you want to validate or the name of the When Condition that you want to apply. d. In the comparator list, select the test that you want to perform on the field or on the When Condition.

©2025 Pegasystems Inc

Choices	Actions
	e. In the value field, enter or select a value to compare to the user input. The `Select values` option lists the fields and values that you can use in the Condition.
The Condition passes when any of the properties meet the criteria	a. Click **Add a row**. 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 or on the When Condition. e. In the value field, enter or select a value to compare to the user input. The Select values option lists the fields and values that you can use in the Condition.

For example:

In a recruitment form, the user must indicate their period of employment at a previous 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:

©2025 Pegasystems Inc

Validation of the employment date range

9. **Optional:** To define more Validation Conditions, repeat steps 5 to 8.
10. Click **Submit**.
11. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
12. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, the application evaluates all Validation Conditions. If the user enters a value that meets any of the Validation Conditions, the respective error message appears. The user can submit the form after they correct the invalid fields.

- Working with Dev Studio forms
- Restricting user input in a field
- Configuring Primary Fields
- Configuring page, page group, and page list properties

PEGA ©2025 Pegasystems Inc

- When Condition Rules
- Validations for ensuring data integrity

Validating field input in complex scenarios

Create Validation Rules to check whether the data values that users enter meet specific criteria, and whether the system can process the information correctly.

While Step Validation supports relatively simple comparisons, Validation Rules help you create more complex operations that include functions and conditional processing. For example, a loan application might require different Validation Conditions depending on the risk rating of the applicant. You can set up a *RiskLevel* property with a value of High, Medium, and Low, and define a set of Validation Conditions for customers that represent High, Medium, and Low risk levels.

> **NOTE:** You define Validation Conditions that check whether the user input in a form is incorrect. If the user input matches the Validation parameters, an error message appears when the user tries to submit the form.

1. In the header of Dev Studio, click **Create menu** > **Process** > **Validate**.
2. On the **New** tab, define the label, context, and Ruleset for the new Validation Rule, and then click **Create and open**.
3. On the **Input** tab, define the input that the application uses to determine which set of Validation Conditions to apply at run time:
   - To define only one Validation set, select **None**.
   - To define a separate Validation set for each value of a property, select **Input property**, and then select the target property.
   - To define a separate Validation set for each Case Status, select **Proposed work status**.
   - To define a separate Validation set for each Flow Action, select **Flow Action Name**.

©2025 Pegasystems Inc

- To define a separate Validation set for each Stage, select **Stages**.

> **For example:**
>
> To set up separate Validation sets for customers with High, Medium, and Low risk levels, select **Input property**, and then enter `RiskLevel` as the target property.

4. On the **Validate** tab, in the **Property** field, enter the property that you want to test.

> **For example:**
>
> To test whether the amount of a down payment is sufficient, in the **Property** field, enter `DownPayment`.

5. In the **Conditions** section, create a Validation set by clicking **Add**.
6. In the **Validation Conditions** window, define the Validation set:
   a. Ensure that the **Enable Conditions** checkbox is selected.
   b. **Optional:** To indicate that the user must fill out a field, select the **Required** checkbox.
   c. **Optional:** To force the system to evaluate subsequent properties even if the current evaluation fails, select the **Continue Validation** checkbox.
   d. In **Select a function**, choose the function that you want to use to evaluate the property.
   e. In the fields below, enter the expression for the function.
   f. In the **Message** field, enter the text you want to display when the Condition is true.
   g. **Optional:** To add an additional Validation function, click **Add a row**, and then select a logical operator to link the functions.
   h. Click **Submit**

©2025 Pegasystems Inc

For example:

To test whether the down payment meets the required minimum, in the **Select a function** list, select **CompareTwoNumbers**. Next, in the fields below, enter `.DownPayment` IS LESS THAN `1000`. In **Message**, enter `Down payment not sufficient`. Now, if the down payment amount is lower than 1,000, an error message appears.

7. If your Validation Rule includes multiple Validation sets for different input values, you can define additional Validation sets:
   a. In the upper right corner of the **Validate** tab, click **Add a new Condition**.
   b. In the **Input** row, enter the input value for which you want to create a Validation set.
   c. Configure the Validation set by completing steps 6.b through 6.h.
   d. **Optional:** To run another Validation set after the current Validation completes, in the **Also execute** row, specify the target Validation set.

For example:

To create a Validation set for high-risk customers, in the **Input** row, enter `High`. Next, add an error message and a Validation set that evaluates if the *.DownPayment* property is lower than 1,500. Now, if a high-risk customer offers a down payment of 1,200, an error message appears, even though that amount would be sufficient for other clients.

Validation sets for two input scenarios

©2025 Pegasystems Inc

8. **Optional:** To run additional Validation Rules after the current Validation completes, in the **Additional Validation** row, select the target Validation Rule.
9. Click **Save**.

What to do next:

- To make the Rule available for the reuse in App Studio, mark it as a relevant record. For more information, see Marking a record as relevant.
- Associate the Validation Rule with a Flow Action. For more information, see Adding a Validation Rule to a Flow Action.

Related tasks

- Validating character patterns
- Configuring Case-wide validation for correct data entry in fields

Adding a Validation Rule to a Flow Action

Ensure that the data that your users provide meets the conditions that a Case requires in order to go forward. By assigning Validation Rules to a Flow Action, you can prevent users from entering information that your application cannot process, and reduce the number of processing errors.

For example, a loan application might require the customer to provide a down payment. You create a Validation Rule that checks whether the down payment is sufficient. Next, you assign the Rule to a Flow Action that executes at the end of the Step in which the customer provides loan information. If the amount of the down payment is too low, the validate Rule prevents the Case from proceeding until the customer increases the down payment.

©2025 Pegasystems Inc

Before you begin:

Create a Validation Rule. For more information, see Validating field input in complex scenarios.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. On the **Workflow** tab, hover over the name of the Process to which you want to add a Validation Rule, and then click **Configure Process** icon.
3. In the top right corner of the screen, click **Open Process**.
4. On the canvas, right-click the connector to which you want to add the Validation Rule, and then click **Open Flow Action**.
5. In the Flow Action Rule, open the **Validation** tab.
6. In the **Validate** field, select the Validation Rule that you want to assign to the connector.
7. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
8. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Related concepts

- Flow Actions

Related tasks

- Configuring a Data Transform as a pre/post processing for a flow action

PEGA ©2025 Pegasystems Inc

Validating character patterns

Check whether the input that your user provides is relevant. By verifying character patterns with edit Validation Rules, you can ensure that the data that users enter into forms matches the field requirements, and reduce the number of processing errors.

For example, if you want your users to enter a valid US postal code in a field, you can use an edit Validation Rule to ensure that the input is consistent with the ZIP Code format. Pega Platform™ includes a wide variety of ready-to-use edit Validation Rules that come with default error messages.

> **Before you begin:**
>
> Create a Validation Rule. For more information, see Validating field input in complex scenarios.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Validate**.
3. Open the Validation Rule that you want to edit.
4. In the **Conditions** section, open a Validation set by clicking **Edit**.
5. In the **Validation Conditions** window, click **Add a row**.
6. In the new row, in the **Select a function** field, select **invokeValidate**.
7. In the **Validation of** field, select the field that you want to check.
8. In the **using** field, select one of the available edit Validation Rules.

   > **For example:**
   >
   > To verify that a string of characters that the user enters in the **ZIP Code** field is a valid US postal code, set *.ZIPCode* as the target field, and *USZipCode* as the Edit Validation Rule.

9. Click **Submit**.
10. Click **Save**.

©2025 Pegasystems Inc

Configuring and tailoring Case Types

Provide users of your application with the ability to manage their work by creating Case Types that best suit their needs. Pega Platform offers a wide range of tools to modify Case Types and customize complex work processes to help users reach their business goals.

For example, by changing a Case Status in the Lifecycle, you can communicate with stakeholders to ensure them that the process moves within the expected time frame.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **Configuring Case Type settings**
- **Configuring complex Processes**
- **Managing Case Types**
- **Automated and intelligent work assignment with Get Next Work**
- **Enabling bulk processing of Cases in Constellation**
- **Changing Case Statuses**
- **Changing a Stage with open Assignments**

Configuring Case Type settings

Better address the individual needs of your customers by configuring Case Types that precisely reflect their business processes. To provide the most efficient solutions for

PEGA ©2025 Pegasystems Inc

your customers, configure their Case Type settings to meet all of their requirements as closely as possible.

When you properly configure Case Type settings, you ensure that users of your application have all the tools that they need to successfully resolve their work. For example, you can define various Case participants and their roles in a Microjourney®. By adding categorized attachments, participants can provide the meaningful data and information necessary to resolve a Case. To save time and improve efficiency, you can also enable automatic Case creation and sharing of data between Cases.

The following figure shows settings categories that you can configure for a Case Type in a Constellation UI application:

©2025 Pegasystems Inc

Settings tab of a Case Type

©2025 Pegasystems Inc

The settings tab in a Traditional UI application additionally includes the Collaboration and the Theme Cosmos UI categories. For more information, see Center-out configuration in Constellation.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **Configuring general Case Type settings**
- **Configuring collaboration settings**
- **Enabling field-level auditing**
- **Sharing data between parent and child Cases**
- **Initializing data at Case creation**
- **Configuring Case creation**
- **Creating a Case by using an API**
- **Defining default data sources**
- **Detecting bias in Case Decisions**
- **Managing concurrent access to a Case**
- **Notifying Case participants about Case events**
- **Predicting Case Outcomes**
- **Defining Case participants**

PEGA ©2025 Pegasystems Inc

- Configuring the Related Cases widget
- Configuring Case-wide validation for correct data entry in fields
- Including indexed data during Case search
- Troubleshooting errors with email instantiation
- Standard Rules used by email instantiation
- Case archival 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.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- Selecting a license category for a Case Type
- Changing the Case icon
- Enabling autosave for forms
- Changing the Case ID format
- Tracking Case workers based on geolocation data
- Providing the Get Next Work link in a Case

PEGA ©2025 Pegasystems Inc

Selecting a license category for a Case Type

Communicate the applicable business use case for your Case Type by selecting a relevant license category. The system uses license categories to report usage of your Case Types, and to help you ensure that you comply with your software requirements. For example, you can select a license category that indicates that most of the Case processing occurs in an internal system.

> **Before you begin:**
>
> To avoid licensing implications and inconsistencies, ensure that you select licensing for your Case Types before you move your application to a production environment.

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 navigation pane, click **General**.
4. In the **License category** list, assign a license to your Case Type:
    - To communicate that the Case manages the processing and completion of work, including updating other transaction systems, select **Work Execution**.
      For example, use this category in the following scenarios:
        - A customer submits a credit card dispute and the dispute is processed and resolved using the software. This is done manually or through straight-through Automation.
        - A company on-boards a new employee and uses the software to get the employee set up in all of the relevant back-end systems.
    - To communicate that the Case routes and tracks work among people or systems, for processing and completion in other transaction systems, and that the Case can be augmented with data from other transaction systems and that customers can submit and update the Case over the web, select **Work Management**.
      For example, use this category in the following scenarios:

©2025 Pegasystems Inc

  - The system routes a new order to an order processing center, and the order is then processed in another system.
  - A customer requests an insurance policy on the client's website and the request is routed to client's staff for processing in another system.
- To communicate that the Case uses the software solely to run an integration with external systems, with no users involved, select **Work Dispatch**.

  Each integration is evidenced by a unique identifier and can be identified within telemetry by the "Work Dispatch" Case typology.

- To communicate that work requires a custom extension for complex scenarios, select **Complex Work Execution**.

The following figure shows the location of the **License category** list:

License category list

5. Click **Save**.

©2025 Pegasystems Inc

Result:

The Case Type has a license category but the default audit reports do not evaluate the classification. For more information about how these classifications affect your license agreement, contact your Pega Account Executive.

- Licenses
- License compliance
- Classifying Rules for license compliance

Changing the Case icon

Customize the icon for each Case Type in your application to help users identify Cases of a given business type 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**.

Choice	Action
Constellation applications	a. In the **Display icon for the Case Type** section, click the current Case icon. **For example:** Click the following default icon:

©2025 Pegasystems Inc

Choice	Action
	Default Case icon setting in Constellation b. In the list of available icons, select the one that best suits your business needs. **For example:** Use the filter option to display briefcase icons, as shown in the following figure: *Filtering icons*
Traditional UI applications	a. Click **Change icon**. b. Click the icon that you want to select for your Case Type, and then click **Done**. **For example:** Select the briefcase symbol for the icon, as shown in the following figure:

©2025 Pegasystems Inc

Choice	Action
	 Selecting an image for the Case icon

3. Click Save.

Result:

The icon that you select overrides the previous icon.

Case icon at run time

4. **Optional:** To remove the Case icon from the Case heading, perform the following steps:
   - To remove a Case icon in a Case Type that you design in a Constellation application, see step 8 in Configuring a Full Page View.
   - To remove a Case icon in a Case Type that you design in a traditional UI application, on the **Settings** tab, click **Theme Cosmos UI**, and then, in the **Case header** section, clear the **Show Case icon** checkbox.

©2025 Pegasystems Inc

- Managing application Cases and data
- Creating Cases from the navigation pane of a Portal
- Changing the Case ID format

Enabling autosave for forms

Prevent users from losing progress during session timeouts by configuring the application to save Assignment forms.

Autosave triggers if the form has unsaved changes and the user is idle for five seconds. This prevents confusion about session activity expectations. For example, if a user fills out a Text Area field in a form and pauses for more than five seconds, the form is saved automatically. However, if the field is a text input field, autosave waits until the user clicks away.

> ⚠ **IMPORTANT:** This feature is not available in Constellation applications. Ensure that you are using Theme Cosmos 05.01 or UI-Kit 15.01. For more information, see Updating Theme Cosmos in your application.

> **Before you begin:**
>
> - Autosave is not run if the Assignment screen does not include a **Save** button or action. To enable autosave functionality, ensure that a save mechanism is present on the form.
> - For single page applications, enable the **Render as a single page** option. For more information, see Adding a dynamic container to a Portal.

Autosave persists data on the server and respects dictionary validations, similar to the **Save for later** action. Note that autosave does not prevent session timeouts due to inactivity.

PEGA ©2025 Pegasystems Inc

Autosave has the following limitations:

Limitation	Description
☐ No Save button or action	Autosave does not trigger if the Assignment screen lacks a **Save** button or action.
☐ Unsupported Portals	Not supported in iframe-based Portals such as Case Manager, Dev Studio, blended UI, and mashup.
☐ Unsupported Harnesses	Not applicable to Create Harness, Assignments in Secondary/Tertiary Ajax Containers, Theme Cosmos Previews, or modal Flow Actions.
☐ Unsupported Clients	Autosave is not supported by Pega Mobile Client™.
Overridden *pyCaseActionArea* Section	If overridden, you must embed *pyAutoSaveBanner* with the same visibility condition and CSS class as the out-of-the-box Section.
Custom popovers	Autosave can cause popovers to remain open and misaligned after saving.
Field focus behavior	Autosave does not trigger if the user is actively typing in a modified field (except Text Area or Rich Text Editor). Focus must move away for autosave to occur.
Session timeout behavior	Autosave persists data on the server and respects dictionary validations, similar to the Save for later action. However, it does not prevent session timeouts due to inactivity.

Enabling the feature at the application level by setting the *pyEnableAssignmentFormAutoSaveFlag* When Rule to `true`.

©2025 Pegasystems Inc

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Decision** category, and then click **When**.
3. Search for and open the *pyEnableAssignmentFormAutoSaveFlag* When Rule.
4. In the first condition row, enter `true`.
5. Click **Save as**, and then save the Rule to your application Ruleset.

> **NOTE:** If you perform a private edit, the autosave functionality becomes available only to you.

Enabling auto-save in a Case Type by selecting the appropriate checkbox in the **General** settings of the Case Type.

6. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
7. On the **Settings** tab, click **General**.
8. In the **Autosave Assignment forms** section, select the **Autosave Assignment form changes regularly when a user isn't typing** checkbox.
9. Click **Save**.

Changing the Case ID format

Change the Case ID format to help Case workers distinguish between instances of different Case Types. Custom Case IDs can help users save time and simplify the organization of work by improving readability. For example, Case IDs of the type Loan Request start with the prefix `LORE-`, and Case IDs of the type New Account start with the prefix `NACC-`.

For each Case, your application generates a unique Case ID in the following format: *prefix-integer*. By default, a Case ID contains the following elements:

©2025 Pegasystems Inc

Prefix

A single character that your application derives from the name of the Case Type that you provide when you create or import a Case Type, and a hyphen. Prefixes are required, but they do not need to be unique.
More than one Case Type can share a prefix.

Integer

A number that increases by one each time that you create a Case. You cannot count the number of Cases in your application based on the integer of a Case ID because the system reserves Case IDs before successfully creating a Case.

The following examples of Case IDs illustrate the different formats that you can use:

- `C-100`
- `JobApp-420`
- `MORT-763`

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **General**.
3. In the **Case ID prefix** field, enter a prefix for your Case ID, for example, `JobApp-`.

> ⚠ **IMPORTANT:** When you enter the prefix, consider the following guidelines:
> - A Case ID prefix can consist of a maximum of 32 characters, including a hyphen at the end. If you do not enter a hyphen at the end of the prefix, the system adds it automatically after you save the settings.
> - Do not use the ampersand (&) and double quotation marks (") in the Case ID.

4. Click **Save**.

©2025 Pegasystems Inc

Result:

The following figure shows a new Case with an ID that starts with the prefix that you configure for the Case Type:

Setting a Case ID prefix

- Creating Cases from the navigation pane of a Portal
- Changing the Case icon
- Managing application Cases and data

Tracking Case workers based on geolocation data

Record the location of users who work on specific assignments by enabling location tracking for Case Types. For example, enable location tracking in tasks for service employees in the field so that dispatchers can more effectively assign work to the employees based on their current location.

©2025 Pegasystems Inc

Before you begin:

Prepare for using the geolocation tracking feature in your application:

- Consult local laws for possible restrictions on location tracking.
- Configure the dynamic system settings with your Google API license key. For more information, see Configuring dynamic system settings for geographic reference.

NOTE: Only standard Pega Platform applications support geolocation tracking. Applications that you build on Constellation do not support this feature. For more information, see Center-out configuration in Constellation.

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. Select the **Enable geolocation tracking** checkbox.
5. Click **Save**.

What to do next:

When you start processing an Assignment, accept a browser prompt to permit location tracking.

If you dismiss the prompt, the geolocation tracking feature might not work. You can display the prompt again by refreshing your browser window.

- Enabling field-level auditing

©2025 Pegasystems Inc

Disabling geolocation tracking

Disable geolocation tracking in Case Types in which it is not necessary to track the location of a Case worker in the audit trail of a Case. For example, you can turn off tracking for all Auto Loan cases because they are processed in the same office.

The standard *pyGeolocationTrackingIsEnabled* When Condition is specialized by class the first time that you enable geolocation tracking for a Case Type. As a result, all audit trail entries include a geographic location. You can disable this feature in the settings for a Case Type or provide a different set of conditions to evaluate.

The system applies each override to an individual Case Type or all Case Types, depending on the position of the *pyGeolocationTrackingIsEnabled* When Condition in the class hierarchy.

> **NOTE:** You can disable geolocation tracking for all Case Types at the same time by deleting all instances of the *pyGeolocationTrackingIsEnabled* When Condition in your application. At run time, the system evaluates the *@baseclass.pyGeolocationTrackingIsEnabled* version, and then always returns a false value.

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** checkbox.
5. Click **Done**.

Related tasks

- Enabling field-level auditing
- Tracking Case workers based on geolocation data

©2025 Pegasystems Inc

- Specializing a Case Type

Providing the Get Next Work link in a Case

Enable the Get Next Work link to display in a Case when no more immediate Assignments are assigned to the Case worker who is processing the Case. The Case worker can then click the link to open the next most urgent Assignment that they are skilled to do.

> **Before you begin:**
>
> Ensure that your application is built with the Constellation UI. For more information, see User experience.

1. In the navigation pane of App Studio, click **Case Types**.
2. In the Case work area, click the **Settings** tab.
3. In the navigation pane, click **General**.
4. In the **Assignment processing** section, select the **Display custom View when no more Assignments are available for the user** and **Enable Get Next Work** checkboxes, as shown in the following figure:

©2025 Pegasystems Inc

Enabling Get Next Work link in a Case UI

> **NOTE:** The Display custom View when no more Assignments are available for the user option adds the out-of-the-box Confirmation View to your Case, which appears when the user completes all of their Assignments in a Case. You can later configure this View to display custom instructions. For more information, see Configuring a confirmation View.

5. Click Save.

©2025 Pegasystems Inc

What to do next:

Learn how to configure and use the Get Next Work functionality at run time. For more information, see Automated and intelligent work assignment with Get Next Work.

Configuring collaboration settings

Complete your Cases more efficiently by enabling collaboration tools such as Pulse messaging, Case tagging, and Case following.

Each setting improves various aspects of collaboration:

- Using tags simplifies and accelerates searching for a particular Case.
- Pulse offers a possibility to discuss work items with collaborators.
- Enabling Case following allows users to track changes in their Cases at run time by receiving notifications.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

Manage configurations for collaboration by performing the following actions:

- **Enabling Case tags**
- **Enabling Case following**
- **Configuring Pulse for Case Types**

©2025 Pegasystems Inc

Enabling Case tags

By enabling users to tag Cases at run time, you speed up Case search. When users use tags to filter Cases, they retrieve relevant information faster.

Users can filter Cases in several ways, for example, by displaying only the recently added or most frequently used tags. To find specific Cases, users can also apply precise tags instead of choosing default filters. For more information, see Filtering tagged Cases.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. Click the **Settings** tab, and then, in the navigation pane, click **Collaboration**.
3. In the **Tags** section, select the **Enable tags** checkbox.
4. Click **Save**.

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.

Enabling Case following

Ensure that users receive notifications about critical updates, such as a change in a Case Status. For example, when a user follows a loan request Case, they receive a notification when their request is approved by the bank.

PEGA ©2025 Pegasystems Inc

> **NOTE:** In Constellation, the application adds the **Followers** widget automatically to the **Utilities** section. For more information, see Configuring a Full Page View.

> **Before you begin:**
>
> Ensure that your application uses traditional UI. For more information, see Application UI setup.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. Click the **Settings** tab, and then, in the navigation pane, click **Collaboration**.
3. In the **Follow** section, select the **Enable follow** checkbox.
4. Click **Save**.

Related tasks

- Following Cases
- Managing Case followers

Configuring Pulse for Case Types

Provide Case workers, 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, such as customer service representatives (CSRs) 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.

PEGA ©2025 Pegasystems Inc

Before you begin:

- Ensure that your application uses a traditional UI architecture or is a UI Kit application with UI Kit 15 or later. For more information, see Application UI setup.

  If you want to configure Pulse for a Constellation application, see Configuring Pulse feed sources for Constellation.

- Create activity feeds to populate user Pulse feeds with notifications. For more information, see Creating feed sources for activity feeds.

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 view, click the **Settings** tab.
3. In the left pane, click the **Collaboration** tab, and then 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 `Internal chat`, as in the following figure: *Setting the* Pulse *gadget label for a case*
Enable posting Pulse messages	Select the **Allow users to add posts** checkbox.

©2025 Pegasystems Inc

Choices	Actions
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 the post type.
Add a feed source	a. In the **Configure additional** Pulse **feeds** section, click **Add Pulse feed**. b. In the **Name** field enter the name of the feed. c. **Optional:** To display the feed source by default, select the **Display Pulse feed by default** checkbox. d. Click **OK**.

4. Click **Save**.

- Preparing for collaboration with users by using Pulse
- Collaborating with users by using Pulse

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.

> **NOTE:** Field-level auditing in Constellation is limited to simple fields, such as Boolean or text fields.

©2025 Pegasystems Inc

Before you begin:

Mark as relevant the properties that are required 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 in the Data Model of your Case. Users of your application can then check previous and current field values, the last user to change the values, and the time of their modification.

For relevant training materials, see the Tracking and auditing changes to data module and the Auditing changes to the value of fields challenge on Pega Academy.

NOTE: If you modify your Cases through background processing or custom activity logic, avoid using the *Obj-Save* method, which can cause field-level auditing to not work correctly. For example, the audit history might display fields as added instead of modified. Instead, use Pega-provided activities, such as *RecalculateAndSave*. For more information, see Obj-Save method.

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** pane, click **Auditing**.
4. In the **Auditing** window, turn on the **Enable field audit** switch.
5. In the **Fields** column, select the checkboxes next to the fields that you want to enable for auditing.

NOTE:

- Field-level auditing does not support Data reference, Page group property, Value Group Property, and Value List Property field types.

©2025 Pegasystems Inc

- Field-level auditing in Constellation is limited to simple fields, such as Booleans or text fields.
- If an application tracks changes in an embedded data field and the scope of the changes is greater than 66%, the application recognizes the changes as delete and add actions, and displays relevant audit information.

6. If you want to enable fields in an Embedded Data field, click the name of the field, and then repeat step 5.
7. Click **Save**.

Result:

- In a traditional UI, when users process the Case, they see the changes to the fields on the **Audit** tab at run time.
- In a Constellation UI, the application now includes a **Field History** View that you can use anywhere in your UI to display changes to the values of tracked fields.

What to do next:

- For Constellation applications, add the **Field History** View to your UI. For more information about managing Views, see Configuring Views.
- 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.

PEGA ©2025 Pegasystems Inc

- **Packaging Rules and data that contain field-level auditing information**
- **The extension point for field-level auditing**
- Configuring Views
- Extending a Data Model for a Case
- Application UI setup

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:

©2025 Pegasystems Inc

- DataAdmin-DataBase-ClassGroup, for example *FLAudit-PegaProjMgmt-Work*
- DataAdmin-DataBase, for example *FLAudit-PegaProjMgmt-Work*

- Packaging your application in a product Rule

The extension point for field-level auditing

You can override the extension point for field-level auditing to include additional information with audit data. For example, you can include the index keys of the audited properties in the audit table.

The extension point that you can use for field-level auditing in Pega Platform version 8.4 and later is the *pySave* activity from the *FLAudit-* class.

> **NOTE:** Field-level auditing extensions are available for traditional UI applications only.

To learn how you can benefit from modifying the extension point for field-level auditing, see the following article with a sample implementation:

- **Displaying the index keys of audited properties in the field-level auditing layout**

Related information

- Tracking and auditing changes to data
- Auditing changes to the value of fields

Displaying the index keys of audited properties in the field-level auditing layout

Increase the readability of the field-level auditing (FLA) layout by adding a column that displays the index keys of the audited properties and the property values that have changed. When you expose the index column inside the Section that displays the

©2025 Pegasystems Inc

property, you can differentiate between the list items more conveniently and view the complete context of the properties that changed.

> **NOTE:** Customization of the FLA layout is possible only for properties of the Page list property type in the traditional UI.

Creating and configuring a property

Create and configure a property to represent the key of audited properties.

1. Create a property with the name *IndexKey* and of the `Text` data type in the *FLAudit-* class in any of the application Rulesets.
2. On the new property form, in the **Additional configuration options** section, select the **Optimize for reporting** and **Optimize for reporting on descendant classes** checkboxes.
   For more information, see Optimizing a property from the user interface and the Optimizing properties for report performance challenge on Pega Academy.

   > **Result:**
   >
   > The *IndexKey* property is now configured to support a new column for presenting key values.

Overriding the pySave extension activity

Override the *pySave* extension activity to get the key index of the Page List properties.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Technical** category, and then click **Activity**.
3. Find and open the *pySave* activity from the *FLAudit-* class.
4. On the activity form, click **Save As**.
5. On the new activity form, in the **Apply to (class)** field, select the class that you created for the Case Type in the *FLAudit-* class.

©2025 Pegasystems Inc

6. In the **Add to Ruleset** field, select the application Ruleset that contains the selected class.
7. Click **Create and open**.

Result:

On the **Pages & classes** tab, you can see the *CurrentFLAPropInfo* page, which is the default context page of the *pySave* extension activity in Pega Platform version 8.8 and later.

NOTE: The *CurrentFLAPropInfo* page provides the context and related information of the property that is being audited. You can use the page to set the index for your audited properties.

The *CurrentFLAPropInfo* page contains the following pages:

pyPropertyChange
: The page that contains the *pyPropertyName* and *pxTrackingIndex* subscripts, which are present only for properties of the Page List type.

pyPropertyInfo
: The page that contains the data of the audited property.

For example, the following figure shows the *CurrentFLAPropInfo* page when the *EmployeeList(1)* property is being audited:

©2025 Pegasystems Inc

Name	Value
pxObjClass	FLAudit-
pzStatus	valid
pyPropertyChange	
	Name / **Value**
	pxObjClass: FLAudit-O8138J-FLAUSTes-Work-FLA
	pxTrackingIndex: 1
	pyPropertyName: EmployeeList
pyPropertyInfo	
	Name / **Value**
	pxObjClass: O8138J-FLAUSTes-Work-FLA
	EmployeeList(1)
	Name / **Value**
	pxObjClass: O8138J-FLAUSTes-Data-Employee
	pyLabel
	EmpID: 1
	EmpName: Ramm
	EmpSkillsDT(1)
	Name / **Value**
	pxObjClass: O8138J-FLAUSTes-Data-Skills
	SkillName: Java
	SkillID: JVa
	pyLabel

Properties on the CurrentFLAPropInfo page

8. On the **Steps** tab, click **Add a Step**.
9. Click **Open Precondition Step**, and then select the **Enable conditions before this Action** checkbox.
10. In the **When** field, enter `CurrentFLAPropInfo.pyPropertyChange.pxTrackingIndex==" "`.
11. In the **if true** field, select **Exit Activity**, and then in the **if false** field, select **Continue Whens**.

Result:

The following When Condition checks whether the audited property is of the Page List type:

©2025 Pegasystems Inc

Checking whether the audited property is of the Page List type

12. Click **Submit**.
13. Depending on the type of index that you want to display for your audited Page List property, configure one of the following settings:

Choice	Action
Configure a numeric index for a Page List property	a. On the **Steps** tab, click **Add a Step**. b. In the **Method** field, select the `Property-Set` method, and then click **Expand to see method parameters**. c. In the **PropertiesName** field, enter the name of the property that stores the index value in the application layer, for example `Primary.PageIndex`, and then, in the **PropertiesValue** field, enter *CurrentFLAPropInfo.pyPropertyChange.pxTrackingIndex*. **Result:** *The index tracking subscript copied into the property that stores the index*
Configure a custom Index for a Page List property	a. On the **Pages & classes** tab, in the **Page name** field, enter the name of

©2025 Pegasystems Inc

Choice	Action
	the page that holds the property, and then in the **class** field, enter the class context for the page. **NOTE:** You can use an existing Pega Clipboard page or create a Pega Clipboard page that holds the property that you want to audit. For example, you can create the `EmployeeListRefPage` page that holds a list of employees as the *Work-.pyEmployeeList* property by entering the name of the page and class to which the page applies. b. On the **Steps** tab, click **Add a Step**. c. Click **Open precondition Step**, and then select the **Enable conditions before this action** checkbox. d. In the **When** field, enter `@equals(CurrentFLAPropInfo.pyPropertyChange.pyPropertyName, "embedded property name")`.

©2025 Pegasystems Inc

Choice	Action
	For example: Enter `@equals(CurrentFLAPropInfo.pyPropertyChange.pyPropertyName,"EmpList")`. e. In the **if true** field, select **Continue Whens**, and then in the **if false** field, select **Jump to Later Step**. f. Click **Submit**. g. On the **Steps** tab, click **Add a Step**. h. In the **Method** field, select the `Page-Copy` method, and then click **Expand to see method parameters**. i. In the **Method Parameters** section, in the **CopyFrom** field, enter `CurrentFLAPropInfo.pyPropertyInfo`. j. In the **CopyInto** field, enter the name of the page that holds your property. **For example:** Enter `EmployeeListRefPage`. k. On the **Steps** tab, click **Add a Step**.

©2025 Pegasystems Inc

Choice	Action
	l. In the **Method** field, select the `Property-Set` method, and then click **Expand to see method parameters**. m. In the **PropertiesName** field, enter the name of the property, for example `Primary.PageIndex`, and then, in the **PropertiesValue** field, enter the property in the page list that you want to display as Index.

14. Click **Save**.
15. **Optional:** To add more Page List properties to the overridden activity, repeat steps 13 through 14.

Overriding a report definition

Override a report definition that derives records from the FLA audit table.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Reports** category, and then click **Report definition**.
3. Find and open the *pyGetRecordsByContext* report definition from the *FLAudit-* class.
4. On the report definition form, click **Save as**.
5. On the new report definition form, in the **Apply to (class)** field, select the class of your Case Type in the *FLAudit-* class.
6. In the **Add to Ruleset** field, select the application Ruleset that contains the selected class.
7. Click **Create and open**.
8. On the **Query** tab, click **Add column**, and then in the **Column source** field, enter `.IndexKey`.
9. In the **Column name** field, enter a name for the column.

©2025 Pegasystems Inc

10. Click **Save**.

> **Result:**
>
> The system adds the *.IndexKey* property to the overridden report definition.

Including the new property in the table with the FLA results

Add the new property in the table Section that displays the FLA records.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **User Interface** category, and then click **Section**.
3. Find and open the *pyFieldGroupHistory* Section from the *@baseClass* class.
4. On the section form, click **Save as**.
5. On the new Section form, in the **Apply to** field, select the class of your Case Type in the *FLAudit-* class.
6. In the **Add to Ruleset** field, select the application Ruleset that contains the selected class.
7. Click **Create and open**.
8. On the **Design** tab, in the table area, click a table column, and then click **Insert column after selected** or **Insert column before selected**.
9. Next to the new column's label area, click **View properties**, and then, in the **Cell properties** pane, in the **Value** field, provide the name for the column. For example, enter `Index key`.
10. Click **Submit**.
11. Next to the new column's field area, click **View properties**, and then, in the **Cell Properties** pane, in the **Property** field, enter `.IndexKey`.
12. Click **Submit**.
13. Find and open the *pyFieldHistory* Section from the *@baseClass* class.
14. Repeat steps 4 through 12.

©2025 Pegasystems Inc

Result:

The new property can be exposed and captured as part of field-level auditing. In the following example, the screen captures the *IndexKey* value and displays the value in the field-level auditing table:

Index key values in the field-level auditing table

Sharing data between parent and child Cases

For greater automation of your business processes, you can share data among the Cases in your Case hierarchy. When you reuse data between parent and child Cases, you increase efficiency, improve data consistency, and as a result, resolve Cases faster.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

Related concepts

- Interacting with parent and child Cases on multiple tabs

©2025 Pegasystems Inc

Aggregating data in a parent Case in Dev Studio

Speed up Case processing by configuring a parent Case to update specific values each time the values in child Cases change. For example, you can create a parent Case that represents a request for an insurance offer, and then supplement the Case with child Cases that visualize life insurance and property insurance in an insurance application. When a customer service representative (CSR) updates the values in the child Cases, the insurance application automatically calculates the total value in the parent Case.

> **Before you begin:**
>
> Create a child Case Type. For more information, see Creating a child Case Type in Case Designer.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. Click the **Settings** tab, and then click **Calculations**.
3. In the **Calculations** section, click **Add calculation**.
4. In the **Calculated property** field, press the Down arrow key, and then select the property in which you want to aggregate data.

   The calculated property stores the sum of other properties, across all child Cases of the parent Case.

   > ⚠ **CAUTION:** Ensure that this property is not associated with a Declare Expression, because the value of the property can be overwritten.

5. In the **Case Type** list, select a class of a Case Type that is a child of your current Case Type.
6. In the **Property** field, press the Down Arrow key, and select the name of a Single-value property property that you want to add to the cumulative property.

PEGA ©2025 Pegasystems Inc

7. **Optional:** To add more properties that the system sums up at run time, click **Add property**, and then repeat steps 5 through 6.

For example:

The following figure shows configuring a property in a parent Case to be a sum of two properties from child Cases:

Configuring data aggregation from child Cases to a parent Case

8. Click **OK**.
9. Click **Save**.

Result:

At run time, an application updates the value of the calculated property in the parent Case each time you change any of the selected properties in a child Case.

- Sharing data between parent and child Cases
- Declare Trigger Rules

©2025 Pegasystems Inc

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 Lifecycle.

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 in Case Designer.

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 **Cases**, and then click the Case Type that you want to open.
2. Click the **Settings** tab, and then click **Data propagation**.
3. In the **Data propagation** section, click a child Case in which you want to reuse data.
4. Click **Add property**.
5. In the **Propagate property value** field, press the Down arrow key, and then select the source property from which to copy data.

©2025 Pegasystems Inc

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, as shown in the following figure:

Entering the full name of a property to propagate to a child Case

6. In the **To property value** field, press the Down arrow key, and then select the name of the property that your application sets in the child Case at run time.
7. Click **OK**.
8. **Optional:** To propagate the values of more properties from the parent Case Type, repeat steps 4 through 7.
9. **Optional:** To apply a Data Transform to set property values:
   a. Select the **Apply Data Transform** checkbox, and then select a Data Transform to apply.
   b. On the **Definition** tab of the Data Transform, in the **Source** field, enter `pyParentPageName`.

> **NOTE:** Do not add `param` before the page property name because the parameter is of the Page Name type.

©2025 Pegasystems Inc

c. On the **Pages & classes** tab, in the **Page Name** field, enter `pyParentPageName`.
d. In the **class** field, enter a class that is equal to the class of the parent Case.
e. On the **Parameters** tab, add a new entry with a name of `pyParentPageName`.
f. In the **Data type** field, select **Page name**.

> **NOTE:** Do not mark the parameter as required because the system provides the value at run time.

10. Click **Save**.

> **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 contextual Cases in traditional UI applications
- Propagating data between parent and child Cases

Initializing data at Case creation

Increase the efficiency of customer service and minimize the need for customer input by initializing data for new Cases. As a result, your application preloads information that you might need at run time, such as location, and uses it to complete the fields when a Case starts.

For example, a Case Type for a mobile application that is available in only one city can use that city as the default location to provide a relevant offer. Loading data automatically for a new Case saves time for developers and customers.

PEGA ©2025 Pegasystems Inc

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

Before you begin:

Ensure that you mark as relevant the fields that you want the application to preload whenever the system creates a new Case. For more information, see Marking a record as relevant.

You initialize data for a Case Type in App Studio by authoring a Data Transform in the Case Type settings. A data transform passes data from one field to another. For each field that you want to prepopulate in the Create Stage whenever you create a Case, you either set the value manually or select another field as the source of the value. To configure more advanced Data Transforms, you can switch to Dev Studio directly from the Data initialization landing page.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **Data initialization**.
3. In the **Data initialization** section, in the **Select field** list, select the field for which you want to set the value at Case creation.

   For example:

   Select **Active Channel**, as shown in the following figure:

©2025 Pegasystems Inc

Selecting a field to prepopulate at Case creation

4. Click the **Choose source type** icon, and then select the source of the value for the field:

Choices	Actions
Manually enter the value for the field	a. Click **Enter text**, as shown in the following figure: *Selecting manual text entry for a field* b. In the **Enter text value** field, enter the text that you want to use as the field value. **For example:** To set the label for all new Cases of the Case Type to the name of a sales campaign, enter `August 2023 Sales`.
Use the value of another field	a. Click **Another field**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **Select field** list, select the field whose value you want to use.

5. **Optional:** To initialize an additional field, click the **Add row** icon, and then repeat steps 3 and 4.
6. Click **Save**.

- **Initializing data at Case creation in Dev Studio**

Related tasks

- Configuring a Data Transform
- Configuring Case Type settings

Initializing data at Case creation in Dev Studio

Automatically load data for a new Case by configuring data initialization for a Case Type. For example, a mobile application uses a customer's location data to provide products that are available in the customer's area. As a result, customer service is more efficient because the customer does not need to manually provide additional information each time they use the app.

> **Before you begin:**
>
> Ensure that you can access Dev Studio. For more information, see Learning about Access Groups.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **Data initialization**.
3. In the **Data initialization** section, click **Configure in Dev Studio**.
4. Configure the Data Transform that you want to run at Case creation.
   For more information, see Data Transforms.

©2025 Pegasystems Inc

5. Click **Save**.

> **Result:**
>
> The system runs the *pyDefault* Data Transform to initialize data whenever you create a Case, and uses this data to populate the fields in the Create Stage.

Related tasks

- Configuring a Data Transform
- Configuring Case Type settings

Configuring Case creation

You can control when child Cases are created by configuring one or more creation methods in a Case Type.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

Use the following techniques to configure the creation methods for each Case Type in your application:

- **Enabling automatic creation of child Cases**
- **Enabling creation of Cases by email**
- Creating an Email Channel

©2025 Pegasystems Inc

Enabling automatic creation of child Cases

Enable automatic creation of a child Case Type each time that a parent Case is created.

> **Before you begin:**
>
> 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.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **Instantiation**.

> **NOTE:** The **Instantiation** tab is available only for child Case Types in Dev Studio.

3. Select the **Automatically by system when** checkbox.
4. 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.

5. **Optional:** To refine the conditions that control whether a Case is created, press the Down Arrow key in the **Allow only when** field, and then select the name of a When Condition.
6. Click **Save**.

©2025 Pegasystems Inc

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 value of true.

- **Enabling automatic conditional creation of child Cases**

- Enabling automatic conditional creation of child Cases
- Enabling creation of Cases by email
- Configuring Case creation

Enabling automatic conditional creation of child Cases

Enable conditional creation of a child Case after creating a parent Case when the Case data meets the conditions that you define.

For example, to save time and automate work, you can configure an application that processes purchase orders to create a shipment Case when the status of its parent Case, which tracks the order and payment, changes to Billing-Approved

.

Before you begin:

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.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **Instantiation**.

©2025 Pegasystems Inc

> **NOTE:** The Instantiation tab is available only for child Case Types in Dev Studio.

3. Select the **Automatically by system when** checkbox.
4. In the **of the below Conditions are met** list, select how many Conditions the Case data needs to meet to create a child Case:
   - To create a child Case when Case data meets any of the Conditions, select **Any**.
   - To create a child Case when Case data meets all of the Conditions, select **All**.
5. Define a dependency by clicking **Add Condition**.
6. In the Case field, press the Down arrow key, and then select a Case Type on which you want your current Case Type to depend.
   This list displays Case Types that share your topmost Case Type and excludes specialized Case Types, descendant Case Types, and Case Types with more than one parent.
7. Select the criteria for creating a child Case:

Choices	Actions
Create a child Case when a parent Case starts	a. In the **Condition** list, select **has started**. b. Click **OK**.
Create a child Case when a parent Case reaches the specified work status	a. In the **Condition** list, select **has work status**. b. Select a Case Status in the list of available statuses. **TIP:** When using work status as a dependency Condition, ensure that your Process has at least

©2025 Pegasystems Inc

Choices	Actions
	one shape that sets the status of a Case to the expected value. c. Click **OK**.
Create a child Case when at least one parent Case completes	a. In the **Condition** list, select **has completed**. b. Click **any** or **all** to indicate how many Cases need to resolve before your application creates a child Case. c. Click **OK**.

8. **Optional:** To enable Case creation only when the specified When Condition evaluates to true, in the **Allow only when** field, press the Down arrow key, and then select the name of a When Condition.
9. Click **Save**.

Result:

At run time, your application creates a new child Case when Case data meets your dependency Conditions, and the When Condition that you provide returns a true value. The limitation of one child Case prevents duplication when Case data fulfills a dependency more than once, such as a parent Case Type reaching a specific Stage.

- Changing Case Statuses
- Configuring Case creation

©2025 Pegasystems Inc

Enabling creation of Cases by email

Associate an email account with a Case Type to provide a way for users to create Cases remotely.

NOTE: The option is not available for child Cases.

Before you begin:

- 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. Configuring outbound email.

   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.

©2025 Pegasystems Inc

> **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 Dev Studio, click **Cases**, 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 Process that runs when the Case is created.

   Although Case Types can have more than one starting Process, an email account is associated with only one starting Process at a time.

> **NOTE:** A starting Process applies only to Case Types without the Create Stage. For more information, see The Create Stage.

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.

©2025 Pegasystems Inc

   e. Select the **Enable email listener** checkbox 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**

Related tasks

- Collaborating with users by using Pulse
- Notifying Case participants about Case events
- Sending automatic emails from Cases

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.

©2025 Pegasystems Inc

- 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.

- Collaborating with users by using Pulse
- Configuring outbound 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.

PEGA ©2025 Pegasystems Inc

TIP: For more examples of API-based communication that is typical of managing a Case Lifecycle across environments, see Calling sequences for DX API endpoints .

1. In the navigation pane of Dev Studio, click **Cases**, 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:

©2025 Pegasystems Inc

Sample code for creating a Case by using an API

5. **Optional:** To see a list of the built-in Pega API REST services in Pega Platform, click View all Pega API.

- POST /cases (Constellation DX API endpoint)
- POST /cases (traditional UI DX API endpoint)
- Embedding Constellation UI in web pages
- Creating Cases with a web embed

Defining default data sources

Provide explicit sources of information for your Cases by defining default Data Pages that an application uses to retrieve Case information.

When you expand your application by creating objects that interact with your Case Type or data object, such as Case references, those objects load data from the Data Pages that you set as default. As a result, data management in your application is more transparent and easier to analyze.

©2025 Pegasystems Inc

For example, you can define *Contact information* as a default Data Page for the **Expand account** Case Type. Then, when you create objects that interact with this Case Type, the system prompts the *Contact information* Data Page as a default data source for the new object.

> **Before you begin:**
>
> Create a Data Page that stores information in a structure that meets your business requirements. For more information, see Creating a Data Page.

1. Navigate to the element of your application for which you want to define a default data source:

Choices	Actions
Add default data sources to a Case Type	a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open. b. In the Case working area, click the **Settings** tab. c. In the settings pane, click **Default data sources**.
Add default data sources to a data object	a. In the navigation pane of App Studio, click **Data**. b. In the **Data objects** section, click the data object for which you want to define default data sources. c. Click the **Settings** tab, and then in the settings pane, click **Default data sources**.

2. Define the sources that your application uses to fetch information:
   - To define a default Data Page that loads information, in the **Default record lookup Data Page** list, select the Data Page that you want to use, and then in the **Parameters** section, enter the parameters that you want to pass to the Data Page.

©2025 Pegasystems Inc

You can select only Data Pages with the Page structure.

- To define a default Data Page that loads information in the form of a list, such as a list of office locations that are available for users, in the **Default list Data Page** list, select a Data Page that you want to use.

  You can select only Data Pages with a List structure.

- To define a savable Data Page that loads information and that you can use to update and save Case information, in the **Default save Data Page** list, select the Data Page that you want to use, and then, in the **Parameters** section, enter the parameters that you want to pass to the Data Page.

  You can select only savable Data Pages.

3. Click **Save**.

Result:

At run time, an application sources information from a default Data Page that you provide. Additionally, when you create a Case reference, an application uses a default Data Page of a target Case as a source of information. For more information, see Referencing a Case Type.

For example:

The following figure shows `Expand account` Case Type settings with `Contact information` as a default Data Page. The figure also shows a dialog box with the configuration to create a new field in an `Offer opportunity` Case Type. The field is of a Case reference type and references the `Expand account` Case Type. Because `Contact information` is a default Data Page for the `Expand account` Case Type, the Case reference uses this Data Page as a source of information by default.

©2025 Pegasystems Inc

Default data sources

- Data Pages overview
- Savable data pages
- Extending a Data Model for a Case
- Saving data in a Data Page as part of a Case Lifecycle
- Preloading a Data Page

©2025 Pegasystems Inc

Detecting bias in Case Decisions

Ensure fair outcomes by monitoring Case Decisions for unwanted bias. Define the Decisions and fields that you want to check for bias and get regular reports on Decisions that fall outside the allowed ranges for bias. Improve your Decisions to ensure compliance with ethical standards and regulations.

> **NOTE:** Detecting bias in Case Decision strategy is a feature of self-optimized workflows in Pega Process AI™.

The Ethical Bias check analyzes Decision steps in the Case workflow to measure and report bias in protected fields. Regular analysis helps you to identify violations and gain insights for when you configure Decisions to reduce bias. This ensures fair outcomes for all stakeholders.

For example, a Case Type for processing job applications includes a Decision to automatically approve or reject candidates to proceed to the interview stage. Regulations require that this Decision does not discriminate against female applicants. You can comply with the regulations by monitoring the *Triage job application* Decision for bias in the gender field.

PEGA ©2025 Pegasystems Inc

A Case Type for processing job applications that includes Decisions

The *pyRunEthicalBiasCheck* job scheduler runs daily to perform the ethical bias check and generate the **Ethical bias detected** Insight on the Explore Data landing page. The system stores the results of the ethical bias analysis in the *Data-Decision-BiasResults* class in the *pr_data_decision_biasresults* database table. For more information about the underlying algorithms, see Ethical Bias check algorithms

Before you begin:

- In the header of Dev Studio, click **Configure** > **System** > **Release** > **Toggles**, and then toggle on the **EnableBiasDetection** switch for bias detection. For more information, see Toggling features on and off.
- Ensure that a file repository is set up in your system. The system uses the repository to record the results of each Decision and the values of the protected fields. The ethical bias check runs on a part of this data.

©2025 Pegasystems Inc

Pega Cloud® **systems**

Pega Cloud® systems store Decisions in the *pegacloudfilestorage* repository.

Client-managed cloud and on-premises systems

To create a repository, your access group must include the *PegaRULES:RepositoryAdministrator* access role. For more information, see Creating a repository.

- Set the **EthicalBiasRepository** dynamic system setting to the name of your repository.

Enabling ethical bias detection consists of the following tasks:

1. Recording Decision results
2. **Optional:** Defining the ethical bias window
3. Defining the ethical bias policy in the Case Type settings

Related concepts

- Explore Data

Recording Decision results

In the process flow for the Decision step that you want to monitor, for each connector that leads from the Decision shape, add a Utility shape that calls the *pxStoreCaseDecisionsForEthicalBias* activity, which records the Decision results with the values of the protected fields to the file repository.

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 the Decision step that you want to monitor for ethical bias.
3. In the **Step** pane on the right, click **Open in Dev Studio**.

©2025 Pegasystems Inc

4. In the process flow, for each connector that leads from the Decision shape, add a Utility shape, and then define it with the following parameters:
   a. In the **Rule** list, select the **pxStoreCaseDecisionsForEthicalBias** activity.
   b. In the **DecisionID** field, enter the name of the Decision shape.
   c. In the **DecisionOutcome** field, enter the outcome of the Decision that you want to log.
      The outcome should be the name of the connector for this Utility shape.
   d. Click **Submit**.

Configuration of a Utility shape for ethical bias

5. In the upper-right corner of the **Edit Flow:** *<flow_name>* tab, click **Save**.

©2025 Pegasystems Inc

For example:

The following figure shows an example of a process flow that contains a Decision shape with two connectors that lead out of it. Each connector has a Utility shape with parameters for recording data for ethical bias:

An example of a process flow with Utility shapes on the connectors that follow a Decision shape

Defining the ethical bias window

The Ethical Bias check evaluates Decision results from the past 30 days to identify potential biases. You can adjust the lookup period by modifying the *BiasCheckWindow* dynamic system setting.

For more information on how to change the values of dynamic system settings, see Configuring dynamic system settings.

Defining the ethical bias policy in the Case Type settings

Enhance transparency and fairness in decision-making by enabling automated bias detection in Case Type settings. Monitor specific Decisions and fields for bias, set acceptable bias levels, and receive real-time insights if bias exceeds the allowed range.

©2025 Pegasystems Inc

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** pane, click **Ethical bias**.
4. In the **Ethical bias** section, select the **Enable bias detection** checkbox.
5. In the **Decisions** section, click **Add Decision**.
6. In the **Add Decision** window, in the **Decision** list, select the Decision that you want to check for bias.
7. Select the positive and negative result, and then click **Add**.

The Triage job application Decision with the positive and negative results

Result:

The list in the **Decisions** section displays the Decision and results that you selected.

> **NOTE:** At run time, the system logs the Case Decisions in the repository destination that you can see in the helper text in the **Decisions** section, as shown in this figure:

©2025 Pegasystems Inc

A Decision selected for the Ethical Bias check

8. Repeat step 5 through step 7 to add more Decisions.
9. In the **Fields** section, click Add field.
10. In the **Add field** window, in the Field list, select a field that you want to check for bias.

    In the following example, the allowed bias for the gender field is in the moderate range:

©2025 Pegasystems Inc

Allowed bias for the Gender field

Result:

The system automatically determines the bias measure of a field based on its field type. The bias measure can be numeric (measured as a Gini coefficient), or categorical (measured as disparate impact). You can change the value if needed, for example when an integer is used for categorical values instead of numbers. For example, an integer field might contain the values 1, 2, 3, and 4, which represent credit risk statuses: green, blue, yellow, and red. In this case, you can specify that the field is actually a categorical bias field.

11. In the **Allowed bias** list, select the allowed level of bias, for example, Minimal. If the detected bias falls outside the selected range, the algorithm identifies the bias in that field as unwanted.
12. Click Add.
13. Repeat step 9 through step 12 to add more protected fields.

Result:

The list in the **Fields** section displays the details of the fields that you added and their allowed level of bias, as in this figure:

Fields selected for the Ethical bias check

14. In the upper-right corner of the **Ethical bias** window, click Save.

©2025 Pegasystems Inc

Result:

The Ethical Bias check runs based on the Decisions and protected fields that you selected for monitoring.

15. **Optional:** To see the Insight, in the navigation pane of App Studio, click **Explore Data** > **Bias Results**, and then click **Ethical bias detected**.

Result:

The Insight lists the Decisions and protected fields for which the algorithm detected bias that exceeds the allowed range, as in the following figure:

Unwanted bias found	Case Type	Decision	Field	Value	Bias measure	Allowed bias	Detected bias	Confidence interval
True	Insurance claim	Approve reject application	Age	—	Numeric (Gini coefficient)	Minimal (0.00-0.10)	0.30	0.28-0.32
True	Insurance claim	Second round interview	Age	—	Numeric (Gini coefficient)	Minimal (0.00-0.10)	0.40	0.38-0.42
True	Insurance claim	Approve reject application	Gender	X	Categorical (Disparate impact)	Moderate (0.66-1.50)	0.50	0.45-0.55
False	Insurance claim	Approve reject application	Gender	F	Categorical (Disparate impact)	Moderate (0.66-1.50)	1.10	1.09-1.11
False	Insurance claim	Approve reject application	Gender	M	Categorical (Disparate impact)	Moderate (0.66-1.50)	1.20	1.19-1.21
False	Insurance claim	Second round interview	Gender	F	Categorical (Disparate impact)	Moderate (0.66-1.50)	0.90	0.88-0.92
False	Insurance claim	Second round interview	Gender	M	Categorical (Disparate impact)	Moderate (0.66-1.50)	1.00	0.98-1.02
False	Insurance claim	Second round interview	Gender	X	Categorical (Disparate impact)	Moderate (0.66-1.50)	0.80	0.78-0.82

Insight for the Ethical Bias check

Job scheduler for ethical bias

The *pyRunEthicalBiasCheck* job scheduler performs daily ethical bias checks and generates the **Ethical bias detected** Insight on the Explore Data landing page. The system stores the results of the ethical bias analysis in the *Data-Decision-BiasResults* Class in the *pr_data_decision_biasresults* database table.

For more information, see Managing job schedulers.

©2025 Pegasystems Inc

Managing concurrent access to a Case

Improve efficiency of your business processes and ensure that users can bring a Case to a successful resolution by setting a locking strategy for parent Case Types to reduce the risk of losing updates to Cases that users make simultaneously. For example, in an insurance claim review Case, you can lock a Case to ensure that the Case can move to the next Stage only after a customer service representative provides all data necessary for a Case resolution.

> **NOTE:** Only parent Case Types support this option. However, all child Cases inherit locking strategy from a parent Case.

In most configurations, use the default locking strategy to preserve transaction integrity among Cases. Without default locking, you can lose work when other users perform tasks such as bulk processing or escalation actions in a Service-Level Agreement.

You can provide one user or multiple users with concurrent Case access. Ensure that you adjust the locking strategy to your business scenario by considering the following factors:

- To maintain transaction integrity in both parent and child Cases, enable one user to make data updates. For example, the parent Case can contain properties that calculate or sum up values in the child Case. Locking both parent and child Cases at the same time helps to keep the counts or totals in sync.
- If multiple users need to open and review Cases simultaneously without updating the Cases, allow multiple users to access a Case. Allow multiple users to update a Case also in a scenario when a user needs to update a parent Case shortly after updating a child Case. In the background, the system locks a parent Case for a short period of time after updating a child Case, and if only one user can update a Case, the system returns an error when the user updates the parent Case.

In multi-node environments, Case locks have limitations that can affect how users access Cases:

©2025 Pegasystems Inc

- Locks are node-specific, so a Case locked on one node cannot be unlocked from a different node.
- If a node stops working while a user has a Case open, the lock remains active and cannot be released from other nodes.
- The **End Other Session** button does not release locks created on a different node.
- Unlock functions return unsuccessful results when used from a node other than the one where the lock originated.

The system automatically releases locks after the default timeout period of 30 minutes. This timeout period remains the primary method for resolving cross-node locking issues until an enhanced solution becomes available.

To reduce the impact of these limitations, consider adjusting the lock timeout duration to better match your business needs and ensure proper monitoring of node health.

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. In the **Access strategy to open and work on a Case** section, select one of the available locking strategies:

Choices	Actions
Only one user can access a Case	a. Select **Allow one user**. b. **Optional:** To adjust the time-out to your needs, in the **Time out value** *[time-out duration]* **mins.** field, enter how many minutes a lock remains on a Case. The default value is 30.

©2025 Pegasystems Inc

Choices	Actions
	Result: Your application locks the Case for a time-out duration or until the user submits or closes the Case.
Multiple users can access a Case simultaneously	a. Select **Allow multiple users**. Providing multiple users with access to a Case is necessary if you want users to access Cases offline. **Result:** Your application preserves the changes that the first concurrent user makes. All other users who work on the Case receive notifications and must review the changes before they can submit any updates.

4. Click **Save**.

Result:

The locking strategy applies to new and existing Cases.

©2025 Pegasystems Inc

What to do next:

If you have different concurrent-access requirements for parent Case descendants, 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**
- **Locking stand-alone Cases**
- Securing Case access

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 in Case Designer.

PEGA ©2025 Pegasystems Inc

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** checkbox.
Change the default timeout duration	In the **Time out value** *[time-out duration]* **mins.** 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**.

- Case locking
- Managing concurrent access to a Case
- Securing Case access

Locking stand-alone Cases

Ensure data consistency by defining a locking strategy that applies in scenarios when a Case Type is a stand-alone Case, a child Case, or a parent Case. By configuring multiple locking settings at once, you ensure that your Case Type meets access and security requirements in every scenario.

©2025 Pegasystems Inc

For example, in an insurance application, you can define a locking strategy for the Body injury Case Type to define how locking works when the Case Type is a stand-alone Case or a parent for the Review medical records Case Type. At the same time, you can configure locking behavior when the Body injury is a child of a Car accident Case Type. As a result, you save time because you provide configurations for different use Cases at once.

You lock stand-alone Cases by using a Case Type Rule form in Dev Studio.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Case Type**.
3. Open a Case Type that you want to edit.
4. On the **Advanced** tab, in the **Locking** section, define the locking strategy when this Case Type is a parent Case or a stand-alone Case:

Choices	Actions
Only one user can access a Case	a. Select **Allow one user**. b. **Optional:** To adjust the time-out value to your needs, in the **Time out value** *time-out duration* **mins.** field, enter how many minutes a lock remains on a Case. The default value is 30. **Result:** Your application locks the Case for a time-out duration or until the user submits or closes the Case.
Multiple users can access a Case simultaneously	a. Select **Allow multiple users**.

©2025 Pegasystems Inc

Choices	Actions
	Providing multiple users with access to a Case is necessary if you want users to access Cases offline. **Result:** Your application preserves the changes that the first concurrent user makes. All other users who work on the Case receive notifications and must review the changes before they can submit any updates.

5. **Optional:** To leave a parent Case unlocked when this Case Type is a child Case, select the **Do not lock the parent Case when the child Case is being performed** checkbox.
   After resolving a child Case, the system automatically locks a parent Case for a short period of time to update relevant properties. If you leave a parent Case unlocked and another user performs edits in the parent Case when the system attempts to automatically acquire the lock, the error occurs. To prevent errors, allow multiple users to access the parent Case.
6. Click **Save**.

Related concepts

- Understanding Case hierarchy
- Case Types

Related tasks

©2025 Pegasystems Inc

- Securing Case access

Notifying Case participants about Case events

Escalate expired Service-Level Agreements and highlight Cases that require immediate action by sending email and push notifications to Case participants. You can use push notifications for mobile devices, and set up email notifications for all devices that support email. As a result, you ensure that business processes reach resolution within the defined time frame.

For example, a Case worker can receive an email and a push notification after a goal for an Assignment expires, or after an application routes a new Assignment to the user.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

> **Before you begin:**
>
> Ensure that all the Case participants that you need in your business process are defined for the Case Type. For more information, see Defining Case participants.

All Assignments in a Case Lifecycle inherit the notification policy from the Case Type. The following scenarios do not support notifications:

- You set the notification policy after creating an Assignment.
- A user pulls an Assignment from a Work Queue.

  Users can receive notifications after an application routes an Assignment to their Worklists, but not after pulling Assignments from shared Work Queues.

©2025 Pegasystems Inc

- A user transfers ownership of an Assignment to another user.
- An application routes the Assignment to the Worklist of the user who already processes the Case.

> **NOTE:**
>
> When your application sends an email, the system updates the *pxSendDateTime* property on the *pxCorrSummary* page that belongs to *pyWorkPage*. Such updates might cause issues with refreshing a Case at run time. To prevent errors, the *pyEnableCorrSummary* When Rule is set to false by default, and the system does not update the *pxSendDateTime* property. If your business scenario requires updating the *pxSendDateTime* property, in Dev Studio set the *pyEnableCorrSummary* When Rule to true.

- **Sending email notifications about Assignments routed to a user's Worklist**
- **Sending push notifications about Assignments routed to a user's Worklist and missed SLAs**
- **Sending confirmations of completed Assignments**
- **Sending custom notifications about new Assignments**
- **Disabling Assignment notifications**

Related concepts

- When Condition Rules

Related tasks

- Configuring outbound email

PEGA ©2025 Pegasystems Inc

Sending email notifications about Assignments routed to a user's Worklist

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.

> **Before you begin:**
>
> Ensure that all the Case participants that you need in your business process are defined for the Case Type. For more information, see Defining Case participants.

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 navigation pane, click **Notifications**.
4. In the **Email notifications** section, select the **Email user when Assignment is routed to Worklist** checkbox.
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.

©2025 Pegasystems Inc

For example:

```
"The Case "+ .pyID + " 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.

©2025 Pegasystems Inc

Choices	Actions
	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** checkbox. d. Enter the display text and title for the link. e. Click **OK**.

8. Click **Save**.

©2025 Pegasystems Inc

Result:

A user receives an email when an application routes an Assignment to their Worklist.

- Assigning tasks to users
- Assigning users automatically at run time with business logic

Sending push notifications about Assignments routed to a user's Worklist and missed SLAs

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 Engaging mobile users with push notifications, and Configuring a mobile notification delivery 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 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 pane, click **Notifications**.

©2025 Pegasystems Inc

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** checkbox.

   > **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** checkbox.
6. Click **Save**.

- Assigning tasks to users
- Assigning users automatically at run time with business logic
- 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 (CSRs), 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 Lifecycle.

> **Before you begin:**
>
> Ensure that all the Case participants that you need in your business process are defined for the Case Type. For more information, see Defining Case participants.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, in the Process that you want to edit, click **Configure Process**.

©2025 Pegasystems Inc

3. On the 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 Lifecycle, and the 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.

PEGA ©2025 Pegasystems Inc

Before you begin:

Ensure that all the Case participants that you need in your business process are defined for the Case Type. For more information, see Defining Case participants.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, in the Process that you want to edit, click **Configure Process**.
3. On the 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 Rule uses.

- Notifying Case participants about Case events

©2025 Pegasystems Inc

Disabling Assignment notifications

Prioritize the tasks that need the greatest visibility in a Case by disabling notifications for other Assignments.

> **Before you begin:**
>
> Ensure that all the Case participants that you need in your business process are defined for the Case Type. For more information, see Defining Case participants.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, in the Process that you want to edit, click **Configure Process**.
3. On the 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 Case participants about Case events

Predicting Case Outcomes

Analyze Case progress and optimize the way your application handles work by using Predictions in your Case Types. You can add a Prediction to a Case Type to predict various Outcomes, such as the probability of a successful Case resolution or the probability of fraud.

Predicting Outcomes helps you manage your Case work more effectively and achieve the following goals:

- Prioritize work that needs immediate attention.

©2025 Pegasystems Inc

- Prioritize Cases that are most likely to bring you value.
- Route potentially fraudulent Cases for closer inspection.
- Straight-through process Cases that are low risk.

You can use Predictions to enhance any Step of a Case that uses a Condition, for example, in the Decision, Assignment, or Change to Stage Steps. Predictions can be essential when making critical decisions in a workflow.

> **NOTE:** Designing self-optimized workflows by using predictions is a feature of Pega Process AI™.

> **Before you begin:**
>
> - Ensure that decision nodes are present in your system. For more information, see Pega Process AI installation.
> - Create a Prediction that you want to use. For more information, see Creating Predictions for Case management.

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 **Data Model** tab.
3. On the **Data Model** tab, in the upper-right corner, click **Add field**.
4. In the **Field name** field, enter a name for your new field.
5. In the **Type** field, select **Prediction**.
6. In the **Prediction** field, select the Prediction that you want to use.

> **NOTE:** Predictions that you create in Prediction Studio with the Case Type as subject are automatically added to the Data Model of the Case Type, and populate in the **Prediction** list.

©2025 Pegasystems Inc

7. **Optional:** To customize the labels for the output fields that hold the Prediction value, in the **Prediction output fields** section, replace the text in the Probability, Segment, or Value field.

> **NOTE:** Only the Prediction output fields that you configured in Prediction Studio appear in the **Prediction output fields** section.

8. Click Submit.

Result:

The Prediction field appears as a row in the Data Model. The following figure shows an example of creating a Prediction field:

Creating a Prediction field

Related tasks

- Adding a Prediction widget to Theme Cosmos
- Adding a Prediction widget to Constellation

©2025 Pegasystems Inc

- Creating Predictions for Case management

Defining Case participants

Case participants are individuals or organizations involved in a Case who can receive notifications, take action, or simply observe Case progress. You define these participants to reflect real-world roles, such as customers, employees, or external partners, and ensure that the right people are informed or engaged throughout the Case lifecycle.

For example, in a customer complaint Case, you might define the following participants:

- A Customer (external person), to receive status updates.
- A CSR (internal person), to resolve the complaint.
- A Legal Advisor (company party) to review escalated issues.

NOTE:

- In Pega applications, the terms **Case participant**, **Work party**, and **Stakeholder** are often used interchangeably. However, they all refer to individuals or entities involved in a Case - whether actively working on it or simply needing updates about its progress. These participants can be internal users of Pega Platform™ or external parties without direct access to the system.
- As a best practice, use Pega GenAI Blueprint™ to instantly generate optimal Case Type designs - complete with the most suitable Case participants - based on GenAI and Pega's extensive repository of workflow best practices.

  For more information, see Creating a new application from a Blueprint.

In Pega Platform, each Case participant is defined by two attributes:

©2025 Pegasystems Inc

Party role

The functional role the participant plays in the Case, such as **Customer**, **Interested**, or **Owner**. This role reflects the participant's relationship to the Case and the value they derive from it.

Party type

The classification of the participant based on system access. For example, **Person** indicates an external individual without a user ID in the system, while **Operator** refers to an internal user with access to the application.

For more information, see step 6 in Using Data Transforms for automatic creation and prepopulation of Case participants and Technical reference for participant configuration.

In Pega Platform, participant roles are configured in App Studio on the **Settings** > **Participants** tab of a Case Type, as shown in the following figure:

©2025 Pegasystems Inc

Setting for Case participants

For more information, see Adding Case participants to a Case Type.

To streamline role management, you can manage participants in Dev Studio using the Work Parties Rule form, as shown in the following figure:

Work Parties Rule for configuring Case participants

For more information, see Case participants configuration in the Work Parties Rule form.

At run time, the participants in a Case are listed in the **Stakeholders** widget, which you can add to the Case Type:

©2025 Pegasystems Inc

Stakeholders widget with Case participants

For example, the **Stakeholders** widget can display the following participant roles:

Customer
: A party role of the Person type. Represents an external individual who receives value from the Case, such as status updates or resolution outcomes.

Interested
: Also a party role of the Person type. Represents an external individual who follows the Case progress but is not directly impacted by its outcome. Unlike Followers, who must be system users, Interested parties can be external and do not require system access.

Owner
: A party role of the Operator type. By default, the system assigns the person who created the Case as the participant with the Owner role. This default assignment can be updated if needed.

For more information, see Configuring a Full Page View and Adding fields to the Stakeholders widget.

Additionally, when you add or edit participants in Dev Studio, Pega Platform provides the following types of participants that correspond to different classes:

©2025 Pegasystems Inc

Company

A business organization that might be interested in the Case, for example, a third-party vendor or a legal firm in a customer complaint Case.

Government

Government organizations that might be interested in a Case.

Non-profit

Non-profit organizations that might be interested in a Case.

Operator

A user of the application, for example, a CSR who creates the Case or their manager for a customer complaint. The operator type always has access to the application.

Person

An external person without access to the application, for example, a customer in a customer complaint Case.

> **TIP:** When configuring Case participants, consider both the role they play in the workflow and their access level. Use the Work Parties Rule in Dev Studio to assign appropriate party roles and types, ensuring clarity and consistency across your Case Types.
>
> For more information, see Technical reference for participant configuration.

- **Adding Case participants to a Case Type**
- **Using Data Transforms for automatic creation and prepopulation of Case participants**
- **Case participants configuration in the Work Parties Rule form**
- Sending confirmations of completed Assignments
- Sending custom notifications about new Assignments
- Sending email notifications about Assignments routed to a user's Worklist

PEGA ©2025 Pegasystems Inc

- Sending push notifications about Assignments routed to a user's Worklist and missed SLAs

Adding Case participants to a Case Type

Define the individuals, teams, and organizations that receive notifications or interact with a Case by configuring Case participants. These participants can be internal users of your application, such as customer service representatives (CSRs) or managers, or external parties such as customers or vendors.

For example, you can configure a group of CSRs to receive alerts about new Assignments and deadlines, while also including external customers who raised complaints and need status updates.

> **Before you begin:**
>
> Add the **Stakeholders** widget to your application. For more information, see Configuring a Full Page View and Adding fields to the Stakeholders widget.

Adding a new participant role

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 pane, click **Participants**.
4. In the **Participants** section, click **Add participant**.
5. In the configuration window, in the **Role name** field, enter a unique name that indicates the relationship of the participant to the Case Type, for example, `Project manager`.
6. Define role preferences for the participant:

©2025 Pegasystems Inc

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, select either Current user or Reporting manager, and then click OK. **For example:** If you map a participant to a current user, the 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 checkbox.

©2025 Pegasystems Inc

Choices	Actions
	d. **Optional:** To speed up Case creation, enable automatic creation of participants when the Case begins by selecting the **Create participant automatically when the Case starts** checkbox. **NOTE:** If you select the **Role has user account with this application** checkbox during stakeholder configuration in App Studio, the stakeholder is configured as an Operator. In Dev Studio, you can select the stakeholder type manually.
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** checkbox.

©2025 Pegasystems Inc

Configuring advanced settings

7. **Optional:** To gain advanced control over participant roles, data structures, and run-time behavior, click **Open in** Dev Studio, and configure the required settings. For more information, see Case participants configuration in the Work Parties Rule form.

> **For example:**
>
> To initialize or adjust participant data automatically at run time, switch to Dev Studio, and in the Work Parties Rule form, in the **Data Transform** field, define the Data Transform. For more information, see Using Data Transforms for automatic creation and prepopulation of Case participants.

Saving your configuration

8. In the **Participants** configuration window, click **Add**.
9. On the Case Type form, click **Save**.

> **Result:**
>
> The following visualization presents the **Stakeholders** widget at run time:

©2025 Pegasystems Inc

What to do next:

For more information about advanced configuration of Case participants, see Case participants configuration in the Work Parties Rule form, Using Data Transforms for automatic creation and prepopulation of Case participants, and the Case participants and routing topic on Pega Academy.

Related tasks

- Adding fields to the Stakeholders widget

Using Data Transforms for automatic creation and prepopulation of Case participants

Configure Case participants to be automatically created and prepopulated with predefined information during Case initiation by using Data Transforms.

This approach streamlines case creation by automatically populating participant information with predefined values, eliminating the need for manual data entry and ensuring consistent, efficient data capture across your application.

For example, in the **Legal Request** Case Type, you want to automatically assign a legal advisor as a Case participant when a new Case is created. The legal advisor's complete contact information is to be populated automatically without manual input, improving efficiency and reducing data entry errors.

Before you begin:

Add the **Stakeholders** widget to your application. For more information, see Configuring a Full Page View and Adding fields to the Stakeholders widget.

©2025 Pegasystems Inc

1. In the navigation pane of Dev Studio, click **Cases**, 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 pane, click **Participants**.
4. In the **Participants** section, click **Add participant**.
5. In the configuration window, in the **Role name** field, enter a unique name that indicates the relationship of the participant to the Case Type.

   > **For example:**
   >
   > Enter **Legal Advisor**.

6. In the **Type** field, depending on your business scenario, select the role type from one of the following options:
   - If the type of role involves a business organization that might be interested in the Case, for example, a third-party vendor or a legal firm in a customer complaint Case, select **Company**.
   - If the type of role involves a government organization that might be interested in a Case, select **Government**.
   - If the type of role involves a non-profit type organizations that might be interested in a Case, select **Non-profit**.
   - If the type of role involves a user of the application, for example, a CSR who creates the Case or their manager for a customer complaint, select **Operator**.

     The operator type always has access to the application.

   - If the type of role involves an external person without access to the application, for example, a customer in a customer complaint Case, select **Person**.

©2025 Pegasystems Inc

> **For example:**
>
> For the legal advisor role, select **Person**.

For more information about the role types, see Technical reference for participant configurationand Defining Case participants.

7. To the right of the **Data Transform** field, click **Open Data Transform**, and then, in the **Data Transform Record Configuration** form, define the Data Transform by completing the following actions:
   a. Create the new Data Transform record by performing steps 2 through 7 in Configuring a Data Transform.

   > **For example:**
   >
   > In the **Label** field, enter `SetLegalAdvisorPartyRole`, and then in the **Apply to class** field, ensure that the system displays `Data-Party-Person`.

   For more information, see Data Transforms.

   b. In the newly created Data Transform form, on the **Definition** tab, define the following actions that populate participant data based on the listed properties:

Action	Target	Source	Example
Set	*.pyWorkPartyUri*	"*<a unique identifier string that references a specific participant in a Case>*" • For the Person type:	"jordan.keller@lawfirm.com"

©2025 Pegasystems Inc

Action	Target	Source	Example
		pyWorkPartyUri is overwritten with the value of *pyEmail1*. • For the Operator type: pyWorkPartyUri must match the user ID. • For the **Company**, **Government**, and **Non-profit** types: *pyWorkPartyUri* can be any unique identifier, but may be overwritten depending on the role. For more information, see Technical	

©2025 Pegasystems Inc

Action	Target	Source	Example
		reference for participant configuration.	
Set	*.pyFirstName*	"*<participant first name>*"	"Jordan"
Set	*.pyLastName*	"*<participant last name>*"	"Keller"
Set	*.pyResponseMode 1*	"*<notification setting>*" For more information, see Technical reference for participant configuration.	"Email1"

For more information, see Data Transform form - Set action.

c. Depending on the selected notification setting, define a `Set` action that sets the target property with a value, as in the following example for email notifications:

Action	Target	Source	Example
Set	*.pyEmail1*	"*<participant primary email address>*"	"jordan.keller@lawfirm.com"

d. **Optional:** To define more actions for setting additional fields that represent Case participant data, define more `Set` actions, for example, to populate the following typically used fields:

©2025 Pegasystems Inc

Action	Target	Source	Example
Set	*.pyFullName*	"*<participant name>*"	"Jordan Keller"
Set	*.pyTitle*	"*<participant title or salutation>*"	"Senior Counsel"
Set	*.pyCompany*	"*<name of a company or business entity associated with Case participant>*"	"Keller & Associates LLP"
Set	*.pyAddress*	"*<first line of a postal address, typically the street address, for a participant>* "	"456 Constitution Ave"
Set	*.pyCity*	"*<participant city name>*"	"Washington"
Set	*.pyState*	"*<state or province portion of an address>*"	"DC"
Set	*.pyPostalCode*	"*<postal or zip code portion of an address>*"	"20001"
Set	*.pyPhoneNumber*	"*<participant phone number>*"	"+1 202-555-0198"

For more information, see Technical reference for participant configuration.

©2025 Pegasystems Inc

> ⚠ **IMPORTANT:** When configuring Data Transform values, ensure that you follow these critical syntax rules:
>
> - Use quotes for literal values: when setting properties to specific text or email addresses, always enclose the value in quotes - without quotes, the system interprets values as page references, which can cause configuration errors.
> - Validate property references: ensure all target properties, such as *.pyWorkPartyUri*, *.pyFullName*, and others, are valid for the participant data structure.
>
> For more information, see Data Transforms.

8. **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** checkbox.
9. Enable automatic creation of participants when the Case starts by selecting the **Create participant automatically when the Case starts** checkbox.
10. Click **Add**.
11. On the Case Type form, click **Save**.

> **Result:**
>
> When a new Case is created:
>
> 1. The system automatically runs the specified Data Transform during Case initiation.
> 2. The participant role is created with all predefined information populated.
> 3. The participant appears in the **Stakeholders** widget with contact details.

©2025 Pegasystems Inc

Case participants configuration in the Work Parties Rule form

Configure Case participants using the Work Parties Rule form to streamline role management and enable consistent, reusable logic across Case Types - supporting scalable design and reducing maintenance overhead.

NOTE:

- In Pega applications, the terms **Case participant**, **Work party**, and **Stakeholder** are often used interchangeably. However, they all refer to individuals or entities involved in a Case - whether actively working on it or simply needing updates about its progress. These participants can be internal users of Pega Platform™ or external parties without direct access to the system.
- As a best practice, use Pega GenAI Blueprint™ to instantly generate optimal Case Type designs - complete with the most suitable Case participants - based on GenAI and Pega's extensive repository of workflow best practices.

  For more information, see Creating a new application from a Blueprint.

The Work Parties Rule supports repeatable roles, external participants, and advanced mappings, making it ideal for scenarios involving multiple stakeholders, regulatory oversight, or third-party involvement.

The Work Parties Rule form has the two sections: **List of valid parties** and **List parties that may repeat**, as shown in the following figure:

©2025 Pegasystems Inc

Work Parties Rule form

In the **List of valid parties** section, you can configure the following fields:

Field	Description	Example
Party label	The display name for the role. This label is shown in the UI.	For the role Legal Advisor, you might use *LegalAdvisor* as a label so that it is displayed clearly in the UI.
Role	Displays a unique identifier used internally to reference the participant role.	*LegalAdvisor* is used in routing logic or Data Transforms.
Party class	Specifies the class that defines the structure of the party data.	• For a CSR who is an internal user, use *Data-Party-Operator*. • For a customer, an individual not tied to an Operator ID, use *Data-Party-Person*. • For a non-profit organization, use *Data-Party-Org*.

©2025 Pegasystems Inc

Field	Description	Example
		• For a government organization, use *Data-Party-Gov*. For more information, see Defining Case participants.
Party prompt	Provides a short description or tooltip shown in the UI to help users understand the role.	For the Project Manager role, you can enter `Person responsible for overseeing project milestones.`
Data Transform	Optional. Initializes or prepopulates participant data at run time.	Configure and run the *SetProjectManagerDetails* Data Transform to automatically populate the participant's name and contact info based on the Case context. For more information, see Using Data Transforms for automatic creation and prepopulation of Case participants.
VOE?	Indicates whether the participant is visible on entry, which means that the participant is created and	If you select this option, ensure that you configure a Data Transform that adds the participant data to the **Stakeholders** widget when

©2025 Pegasystems Inc

Field	Description	Example
	visible at the beginning of Case processing.	the Case is created at run time. For more information, see Using Data Transforms for automatic creation and prepopulation of Case participants.
Required?	**NOTE:** This setting is currently non-functional and does not enforce any behavior at run time. This setting is slated for removal in a future release. Until then, it is recommended not to rely on it for participant validation. Instead, use validation logic in Processes or Data Transforms to enforce required participants	N/A

©2025 Pegasystems Inc

Field	Description	Example
	where needed. For more information, see Using Data Transforms for automatic creation and prepopulation of Case participants.	

The **List parties that may repeat** section enables you to define roles that support multiple participants. For example, multiple reviewers or stakeholders under the same role. This is useful in the following scenarios:

- Several individuals share the same responsibility; for example, there are multiple reviewers.
- You want to send notifications to a group rather than a single user.

Related concepts

- Defining Case participants

Related tasks

- Adding Case participants to a Case Type

Configuring the Related Cases widget

Help users get a broader perspective on assignments by providing the option to include related Cases in a dedicated widget. For example, in a loan application Case, users can

©2025 Pegasystems Inc

add interrelated Cases about their credit history to ensure that all relevant information is in one place.

Configuring the widget

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 **UX** tab, click the **Full Page View** tab.
3. In the pane on the right side of your work area, in the **Utilities** section, add the widget by clicking **Add** > **Related cases** > **Add**.
4. In the row with the newly added widget, click the **Configure** icon.
5. In the **Edit widget: Related Cases** dialog box, manage the widget settings, and then click **Save**.

   > **For example:**
   >
   > In the **Visibility** list, select **Custom condition** and configure the widget to appear when the assignment status is Open.

Managing related Case Types

6. In the Case Type, click the **Settings** tab.
7. In the menu on the left side of the Case Type, click **Related Case Types**, and then, in the main pane, click **Add related Case Type**.
8. In the drop-down list, select the Case Types that you want to show in the widget.
9. In the upper-right corner of the Case Type, click **Save**.

Related concepts

- Conditional logic in Views

©2025 Pegasystems Inc

Configuring Case-wide validation for correct data entry in fields

Ensure that users input data in the correct format to prevent processing errors during Case creation or updates. To save time, add Case-wide validation criteria instead of validating recurring fields on each individual Assignment.

For example, in a candidate hiring Case Type, you design a View that includes a **Hire date** field. To ensure that users enter the date in a valid format, you add a Validation Rule for the **Hire date** field in the View. However, if you need to configure the same field in multiple Assignments throughout the Case, you can add one Validation Rule for that field that applies to all the **Hire date** fields in the Case. Enforcing a uniform date format for all **Hire date** fields throughout the Case with Case-wide validation streamlines this process by applying consistent Rules across all relevant fields, reducing redundancy and enhancing data integrity.

As a best practice, to ensure consistent data validation across all Channels, implement validation logic at the Case level rather than within the UI layer. This approach guarantees that validations apply universally, whether users interact through web, mobile, conversational, or API Channels. Relying solely on UI-based validations, such as marking fields as mandatory in the View configuration, restricts enforcement to the UI layer. Consequently, when the system processes Cases through APIs or other Channels that bypass the UI, these validations are not enforced, which can lead to data inconsistencies.

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. Select the moment in which you want your application to validate user input:
    - To run validation at Case creation, next to the **When Case is created** option, click **Configure in Dev Studio**.

©2025 Pegasystems Inc

> **IMPORTANT:** Only traditional UI applications and Case Types without the `Create` Stage support validation at Case creation.
>
> If your Case Type includes the `Create` Stage, and you want to configure validation at Case creation, configure validation on the next Stage after the `Create` Stage. For more information, see The Create Stage and Defining the entrance criteria for a Stage.

- To run validation when Case updates, next to the **When Case is saved** option, click **Configure in Dev Studio**.

> **Result:**
>
> The system opens a form in which you create a Validate Rule to undergo evaluation each time users create or save a Case.

4. On the **Create Validate** form, in the **Context** section, in the **Add to Ruleset** field, enter the name and version of an unlocked Ruleset.

> **IMPORTANT:** The default values in the **Label** and **Apply to** fields must remain unchanged so that the Rule resolution can identify the appropriate Rules.

5. Click **Create and open**.

> **Result:**
>
> If you configure validation to occur when users create a Case, the system applies the *Edit Validate: OnAdd* Rule.

©2025 Pegasystems Inc

> If you configure validation to occur when the users save a Case, the system applies the *Edit Validate: Validate* Rule. You complete the Rule forms in the same way.

6. On the **Input** tab, select **Input property**, and then select the target property or click the **Target** icon to define the property.
7. On the **Validate** tab, in the **Property** field, enter the property that you want to test.
8. In the **Conditions** section, click **Add** to create a validation set.
9. In the **Validation Conditions** window, define the validation set:
   a. Ensure that the selection for the **Enable Conditions** checkbox is active.
   b. **Optional:** To indicate that the user must fill out a field, select the **Required** checkbox.
      If you select the **Required** checkbox and users leave the field blank at run time, the system displays the `value cannot be blank` message.
   c. **Optional:** To force the system to evaluate subsequent properties even if the current evaluation fails, select the **Continue Validation** checkbox.
      If you enable the checkbox for multiple properties, users can view all potential validation errors on one page.
   d. In the **Select a function** field, select a Function to validate your property.
   e. In the entry fields that are displayed after you select a function, enter the Expression for the function.
   f. In the **Message** field, enter the text that you want displayed when the condition is true.
   g. **Optional:** To add an additional validation function, click **Add a row**, repeat steps 9.a through 9.f, and then select a logical operator to link the functions.
   h. Click **Submit**.
10. If your Validate Rule includes multiple validation sets for different input values, you can define additional validation sets:
    a. In the upper-right corner of the **Validate** tab, click **Add a new Condition**.
    b. In the **Input** row, enter the input value for which you want to create a validation set.

©2025 Pegasystems Inc

c. Configure the validation set by completing steps 9.a through 9.h.
d. **Optional:** To run another validation set after the current validation completes, in the **Also execute** row, specify the target validation set.

11. **Optional:** To run additional Validate Rules after the current validation completes, in the **Additional Validation** row, select the target Validate Rule.
12. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
13. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a Validate Rule that returns an error message when a user enters a hire date in the future and the run-time effect with the message visible when a user enters a hire date in the future:

©2025 Pegasystems Inc

Validation at design time and run time

What to do next:

To make the Rule available for reuse in App Studio, mark it as a relevant record. For more information, see Marking a record as relevant.

Related concepts

- Validations for ensuring data integrity
- Restricting user input in a field

Including indexed data during Case search

Enable search for a Case Type to ensure that users can find cases based on any text, partial or full, that matches indexed Case data.

Before you begin:

Contact your system administrator to ensure that the *indexing/useDataInstances* dynamic system setting is set to `true`.

1. In the navigation pane of App Studio, click **Cases**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **Search and reporting**.
3. Click **Enable search**.
4. **Optional:** To make reporting faster without putting additional requirements on the database, click **Generate dedicated index**.

Case archival settings

Archiving resolved Cases helps you manage system performance, comply with data retention policies, and reduce database size.

©2025 Pegasystems Inc

For example, to reduce load on the production database and comply with retention policies, an insurance company archives resolved **Claim** Cases 180 days after resolution. The archived Cases remain searchable in read-only mode for audit purposes.

For more information, see Archiving and expunging Case data.

In Pega Platform™, how you configure Case archival depends on the type of application that you use:

- In Constellation applications, you enable Case archival in Dev Studio by updating the *pyShowCaseSetting* When Rule in the *Pega-Case-Settings-ArchivalPolicy* class.

  For more information about configuring the Case archival process in Constellation applications, see Enabling Case archiving in Constellation.

- In traditional UI applications, you configure Case archival in Dev Studio, on the **Settings** tab of a Case Type.

  For more information about configuring the Case archival process in traditional UI applications, see Defining the archiving policy for Case Types.

While the setup process differs, the outcome, which is automated archival of resolved Cases after a defined retention period, is the same.

Configuring complex Processes

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

For example, you can define different paths that a Process can follow after a decision point. You can also create Subprocesses to support additional work that your Case requires.

PEGA ©2025 Pegasystems Inc

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

Typically, you build Processes in Case Lifecycle Designer. For advanced configuration, you can build a Process by adding shapes to the Process diagram in Process Modeler. If you build a Process by using the Process diagram, for future edits you also need to edit the Process by using this tool.

Shapes and connectors

When you work with the Process diagram, you first select Process 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, Process Modeler 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 Process shapes.

PEGA ©2025 Pegasystems Inc

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.

Process diagram

Pega Platform™ conforms to the Business Process Model and Notation (BPMN) standard by using a subset of BPMN, but also extends the standard with smart shapes such as the Capture signature, Cascading approval, and Wait shape. The Process diagram includes a smart shapes palette that provides a set of frequently used, pre-configured utilities and Subprocesses. The shapes in the palette can be used in Process flows with little or no additional configuration beyond what BPMN supports. Pega continues to define and redefine the BPM space by favoring a more robust approach that supports advanced enterprise BPM constructs, such as business rules and Case

©2025 Pegasystems Inc

Management (for example, treating GUI calls as a service instead of as an intrinsic part of the BPM flow).

- **Creating a stand-alone Process**
- **Changing the path of a Process**
- **Subprocesses**
- **Flow Actions**

Related tasks

- Processes in a Case Lifecycle
- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools

Creating a stand-alone Process

Create a stand-alone Process to support or supplement the Steps in a Case Lifecycle. By creating a Process outside the context of a Case Lifecycle, 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 Process diagram. If you want to use a Process 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:
   a. In the header of Dev Studio, click **Create menu** > **Process** > **Flow**.

©2025 Pegasystems Inc

b. In the **Label** field, briefly describe the purpose of the Process.

> **For example:**
>
> Enter `Collect feedback`.

c. **Optional:** To change the default identifier for the Process, click **Edit**, and then provide a unique value in the **Identifier** field.
d. In the **Context** section, select an application layer to store the Process.
e. In the **Apply to (class)** field, click the Down arrow key, and then select a class that stores the Process.
f. In the **Add to Ruleset** list, select a Ruleset and a Ruleset version to store the Process.
g. Click **Create and open**.

> **Result:**
>
> The Process 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*

2. On the **Diagram** tab, click **Flow Shapes**, and then select a shape that you want to add to the Process diagram.

©2025 Pegasystems Inc

For more information about flow shapes, see Process shapes.

3. Connect the shapes by dragging connectors from one shape to another.

> **TIP:** Connectors that work properly are green. If you cannot place a connector in a selected place, the connector is red.

4. Click Save.

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.
- To make the Process available for reuse in App Studio, mark it as a relevant record. As a result, you can conveniently use the Process while building Case Types in App Studio. For more information, see Marking a record as relevant.

  Processes can be reused in App Studio as smart shapes, encapsulating advanced and complex functionalities in a low-code element that developers and business representatives can incorporate into their Case Types in App Studio. For more information about Studios interoperability and modular reuse, see Reusable elements in App Studio.

- **Process shapes**

©2025 Pegasystems Inc

- **Types of Processes**
- **Assignment shapes in Processes**
- **Annotating a Process**
- **Keyboard shortcuts for Processes**
- **Creating a Screen Flow in Dev Studio**

Related concepts

- Flow Actions
- Intelligent routing

Related reference

- Activity form - how to create activities for Processes

Process shapes

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

> **NOTE:** Process shapes of the Automations category are available only for Processes that belong to a *Work-* class and its subclasses. If you configure a Process that belongs to a *Data-* or *Rule-* class, the number of available Automation shapes is restricted.

Name	Description	Image
Annotation	Adds notes or comments to the Process without affecting how the	

©2025 Pegasystems Inc

Name	Description	Image
	Process runs. For example, in a Case of preparing a business offer for a potential business partner, Case workers can add comments about possible modifications of the options that the offer includes. For more information, see Annotating a Process.	
Ask a question (Conversation)	Asks users a question in a button, quick reply, yes/no, or open format, and saves the answer in a Case or system property. The shape can also use smart Validation to ensure that the user response is in the correct format, for example, a date format. You can only add this shape to a conversation Process. For example, in a conversation Process related to a vehicle ride request cancellation fee dispute, you can configure the shape to ask the reason for disputing the cancellation charge using buttons with three choices: specifying that the ride never arrived, the ride was late, or another reason. For more information, see Adding a Case Type conversation process for a conversational Channel.	

©2025 Pegasystems Inc

Name	Description	Image
Assign to robot queue	Identifies the robotic Work Queue and Automation to run on the Work Queue for Process Automations. As a result, you promote automation and minimize human engagement in the Processes. For example, a robotic queue can automatically check currency rates for a banking application during international money transfers. For more information, see Assigning a task to a robotic Work Queue. **NOTE:** The Assign to robot queue shape is an advanced shape available only in Dev Studio.	
Assignment	Creates a task in a Work Queue or Worklist so that a user can provide input to the process. For example, you can collect income information and banking history from a customer in a loan request review process. For more information, see Collecting information from a user.	

©2025 Pegasystems Inc

Name	Description	Image
Assignment Service	Identifies an activity that connects your instance of Pega Platform to an external system to send an Assignment for external processing. The Assignment Service shape moves a Case to the next Step only after receiving the response from the external system when the Assignment processing is complete. For example, an application can route an Assignment to a third-party evaluator to estimate the value of vehicle damage in an insurance claim. For more information, see Assigning a task to an external application. **NOTE:** The Assignment Service shape is an advanced shape available only in Dev Studio.	
Conversation flow (Conversation)	Starts another existing conversation Process within the context of the work item, or in an embedded page if you create a nested Data Model for your Case using the object-oriented	

©2025 Pegasystems Inc

Name	Description	Image
	approach. For example, for a vehicle repair Case, you can configure the system to start an additional conversation Process within the context of the work item that only asks users a series of questions about the extent of their vehicle damage. For more information, see Adding a Case Type conversation process for a conversational Channel.	
Custom event (Conversation)	Enables customers to interact with a website or run a custom client-side event during a chat conversation, either synchronously or asynchronously. For example, you can let customers download a file from a website before returning to the chat conversation. For more information, see Adding a Case Type conversation process for a conversational Channel.	
Decision	Evaluates an Expression or calls a Rule, such as a decision tree, to determine which Step is next in the process progression. For example, in a loan request process, the Case might follow multiple paths depending on the loan amount. see Adding Decisions to Processes.	

©2025 Pegasystems Inc

Name	Description	Image
End	Designates the end of the Process.	
Integrator	Identifies an activity that connects your instance of Pega Platform to an external system to send or receive data. The Integrator shape runs synchronously and moves a Case to the next Step after it receives the response from the external system, typically after a few seconds. For example, you can use the Integrator shape to retrieve customer account balances or verify account numbers in an external database. For more information, see Connecting a Process to an external application. **NOTE:** The Integrator shape is an advanced shape available only in Dev Studio.	
Question (Automation)	Asks users a question in a format that your business process requires, such as plain text, a picklist, or a slider, and saves the answer in a Case or system	

©2025 Pegasystems Inc

Name	Description	Image
	property. As a result, you can selectively capture user input without the formal structure of a questionnaire. For example, in an insurance process, a Case worker asks a customer about the type of insurance claim that they want to make. For more information, see Asking a question in a Case. **NOTE:** The Question shape is available only in Dev Studio.	
Question page (Automation)	Asks users a group of related questions in various formats that your business process requires, such as plain text, a picklist, or a slider, and saves the answers in Case of system properties. For example, in a sales call, a telemarketer asks a customer a set of questions about their hobbies, and then uses the answers to provide only relevant products. For more information, see Asking a group of related questions in a Case.	

©2025 Pegasystems Inc

Name	Description	Image
	NOTE: The Question page shape is available only in Dev Studio.	
Questionnaire (Automation)	Runs a questionnaire child Case that asks users a set of questions in various formats. For example, you can provide a text box in which users can enter their answers directly, or create a radio button matrix, so that users can select their answers from a group of radio buttons. For more information, see Running a questionnaire in a Case.	
Send a message (Conversation)	Displays a text message to the user. You can only add this shape to a conversation Process. For example, in a conversation Process related to a vehicle ride request cancellation fee dispute, you can configure the shape to inform users that a charged cancellation fee will be refunded within the next few days. For more information, see Adding a Case Type conversation process for a conversational Channel.	

©2025 Pegasystems Inc

Name	Description	Image
Split for Each	Creates multiple instances of the same Subprocess that later rejoin the parent Process. Users can work on the Subprocesses simultaneously, which accelerates Case resolution. For example, during the recruitment process of a job candidate, a hiring manager and an HR worker can review and assess the results of a job interview in parallel. For more information, see Running multiple instances of the same Subprocess with a Split for Each shape. **NOTE:** The Split for Each shape is an advanced shape available only in Dev Studio.	
Split Join	Creates multiple Subprocesses that later rejoin the parent Process. Users can work on the Subprocesses in parallel, which saves time during Case resolution. For example, during the recruitment of a job candidate, an HR worker can conduct the background check while the hiring manager reviews and assesses the	

©2025 Pegasystems Inc

Name	Description	Image
	job history and certificates. For more information, see Running multiple instances of different Subprocesses with a Split Join shape. **NOTE:** The Split Join shape is an advanced shape available only in Dev Studio.	
Start	Designates the start of the Process. For more information about the advanced configuration options, see steps from 3 to 11 in Creating a Screen Flow in Dev Studio.	
Subprocess	Calls another Process from the current Process. For example, in a hiring process, you can add a Subprocess of running a background check. For more information, see Subprocesses.	
Swimlane	Identifies a group of related shapes that are associated with the same organizational unit or business purpose. For example, after accepting a job candidate, an HR worker collects the necessary	

©2025 Pegasystems Inc

Name	Description	Image
	documents and prepares a legal agreement, while an IT worker prepares a work station and configures a laptop for the new worker. You can use a swimlane to categorize the Assignments between the HR worker and the IT worker, which facilitates routing of work. **NOTE:** The Swimlane shape is an advanced shape available only in Dev Studio.	
Utility	Calls an activity or an Automation. By using the Utility shape, you can perform processing, such as computations or data retrieval, without human input. For more information, see Calling an activity or Automation from a Process. **NOTE:** The Utility shape is available only in Dev Studio.	

©2025 Pegasystems Inc

Name	Description	Image
Automations		
Approval	Routes a Case to one or more reviewers for an approval, based on a user name, reporting structure, or authority matrix. For example, in a hiring process, to hire a candidate for a managerial position, a Case might require an approval from multiple managers on different levels. For more information, see Requesting approval from users.	
Attach content	Attaches a file or an URL to a Case. You can select different types of files to attach. For example, in a hiring process, you can attach scanned university diplomas of a job candidate. For more information, see Attaching content to a Case.	
Change to a Stage	Moves a Case to a specific Stage that you select. For example, in a hiring process, you can move a Case to an optional Stage for an additional interview. For more information, see Automatically moving a Case to a different Stage.	

©2025 Pegasystems Inc

Name	Description	Image
Change to next Stage	Moves a Case to the next Stage. For example, in a hiring Process, after a job candidate provides necessary documents, a Stage moves to the next Stage in which a human resources worker reviews the documents. By default, a Case moves to the next Stage after all the Assignments in a current Stage are complete. Use this shape when you want to move the Case before completing some Assignments. For more information, see Stages in a Case Lifecycle.	
Change to Previous Stage	Moves a Case to the previous Stage. For example, in a hiring process, after a job candidate's documents are re-verified in an Alternate Stage, Case processing returns to the last visited Primary Stage to continue the regular path of the hiring process. For more information, see Automatically moving a Case to a different Stage.	
Connect Generative AI	Calls a Generative AI Rule to collect data specified in the Rule form. For more information, see Populating Case data with personalized AI-generated information.	

©2025 Pegasystems Inc

Name	Description	Image
Create Case	Creates a top-level Case or one or more child Cases. For example, in a car accident insurance request review, you can create a child Case to review vehicle damage. For more information, see Starting child and Remote Cases from the main Case context and Creating contextual Cases in traditional UI applications.	
Create PDF	Creates a PDF file from a specified section and attaches the file to the Case. For example, in a car accident review request, you can create a PDF file with information about a vehicle damage. For more information, see Creating a PDF from Case data.	
Generate document	Generates a document that stores information from a Case. You can select which fields that correspond with Case data you want to include in the documents. For example, in a hiring Process, you can generate a document that stores personal details of a candidate, the interview score, and a decision about accepting a candidate. For more information, see Generating Case documents.	

©2025 Pegasystems Inc

Name	Description	Image
Load Data Page	Preloads a Data Page with information that your application displays to users. As a result, Case processing is faster, and users avoid waiting for an application to load data on demand. For example, in an onboarding Case, an application can preload a Data Page that stores information about equipment that a new hire requests before a worker who processes the Case needs to view the information. For more information, see Preloading a Data Page. **NOTE:** The Load Data Page shape is available only in Dev Studio.	
Save Data Page	Saves Case information as a separate Data Page so that you can reuse data outside of a Case. The information needs to come from a savable Data Page or an autopopulated property that points to the savable Data Page. After an update, an application saves the updated information to the same specified Data Page. For	

©2025 Pegasystems Inc

Name	Description	Image
	example, in an onboarding Case, you can store information about a new hire, such as an office location or requested equipment, as a Data Page, and then reuse this information in other places of your application that require a Data Page. For more information, see Saving data in a Data Page as part of a Case Lifecycle. **NOTE:** The Save Data Page shape is available only in Dev Studio.	
Post to Pulse	Creates a message that is sent to the Pulse social stream. For example, when a customer service representative (CSR) collects additional details from a customer, they can post a message to inform other stakeholders that the Case involves. For more information, see Posting messages to a Case.	
Push notification	Sends a notification to an iOS or Android mobile device to indicate that a Case requires a user action. For example, when a hiring process moves to a Stage where a	

©2025 Pegasystems Inc

Name	Description	Image
	hiring manager needs to accept or reject a candidate, the manager receives a notification. For more information, see Sending push notifications from Cases.	
Run Data Transform	Calls a Data Transform to convert data from one format and class to another format and class. For example, while processing a purchase order, you can populate the shipping address with the data that the user provides as their billing address. For more information, see Populating Case data with a Data Transform.	
Search duplicate Cases	Returns a list of Cases that are similar to a current Case based on criteria that you provide. For example, if a customer creates a loan request similar to another loan request from the same customer that is still in progress, you can find and then close the second request. For more information, see Searching duplicate Cases.	
Send email	Sends an email to one or more work parties. For example, in a hiring process, a job candidate can receive an email with a summary	

©2025 Pegasystems Inc

Name	Description	Image
	of the process. For more information, see Sending automatic emails from Cases.	
Send notification	Sends a notification to users over one or more channels based on the configurations that you define. For example, in a loan request review, you can notify a customer when a Case moves forward and changes its status. For more information, see Sending event notifications from Cases.	
Update a Case	Updates a selected Case or all child Cases with details that you select. For example, at a certain point of a loan request review, you can update the Case Urgency to a higher value to ensure that the Case meets the resolution on time. For more information, see Propagating data between parent and child Cases.	
Wait	Pauses an Assignment for a specified length of time or until one or more Cases reach a specific status. For example, in a hiring process, you can pause the Case until a human resources worker finishes a background check of a candidate. For more information,	

©2025 Pegasystems Inc

Name	Description	Image
	see Pausing and resuming Processes in Cases.	

- **Connecting shapes in a Process**
- **Shape icons in a Process**

- Case Lifecycle elements
- Creating a stand-alone Process
- Running multiple instances of different Subprocesses with a Split Join shape
- Running multiple instances of the same Subprocess with a Split for Each shape
- Activity form - how to create activities for Processes
- Intelligent routing
- Flow Actions

Connecting shapes in a Process

Use connectors to integrate shapes in a Process. By using connectors, you can define the different paths that a Process can take at run time.

> **NOTE:**
>
> - You cannot make connections to Annotations or Swimlane in a Process.
> - You cannot configure connectors in a Screen Flow because this type of Process follows one path only. The Create Stages use Screen Flows.
>
> For more information, see Process shapes.

1. Add a new shape to a Process:
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. On the **Workflow** tab, in the **Case Lifecycle** section, click a Process.

©2025 Pegasystems Inc

   c. In the Process property pane, on the **General** tab, click **Open Process**.
   d. On the toolbar, click **Flow Shapes**, and then click a shape.
2. On the graphical canvas, hover over your shape, and then drag a connection point from the shape to the connection point of another shape in the Process.
3. Double-click the connector arrow that leads from your shape to open the **Connector properties** pane, and then configure the connector based on the type of shape that you are adding to your Process:

Choice	Action
Assignment or Assignment Service shape	a. In the **Flow Action** field, select a Flow Action that displays instructions to the user who performs the Assignment. b. In the **Likelihood** field, enter a number between 1 and 100 that indicates the percentage of time that the user performs the flow action relative to the flow actions of other outgoing connectors from the Assignment. If you are providing only one connector for the Assignment, enter a likelihood value of 100.
Any other shape	a. In the **Condition type** list, select an option that defines when the Process follows the connector at run time. **NOTE:** You can use the following options:

©2025 Pegasystems Inc

Choice	Action
	Always Follows the connector always because there is only one path to follow. **Else** Follows the connector when no other connector meets the criteria that you define. **Result** Follows the connector when the return value of the shape equals the value that you provide. **When** Follows the connector when the Boolean Expression or When Condition that you provide returns a true value. **Percentage** Follows the connector for a specific percentage of Process runs. For example, you can ensure that 20 percent of mortgage

©2025 Pegasystems Inc

Choice	Action
	applications take a different path in the Process for auditing purposes. **NOTE:** This option is available only when you connect a Decision shape that is a **Fork** type. b. In the **Result** field, enter a return value to evaluate at run time. This value can be text or a predefined value, based on the configuration of the shape 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.

©2025 Pegasystems Inc

Choice	Action
	NOTE: This field is available only when you set the connector type to Result. c. In the **When** field, enter a When Condition to evaluate at run time. **NOTE:** This field is available only when you set the connector type to When. d. In the **Likelihood** field, enter a number between 1 and 100 that indicates the expected percentage of time that your Process follows this connector at run time. Your application evaluates connectors in order of decreasing likelihood and selects the first connector that returns a true value.

4. To display a message in the audit trail of the work item, in the **Audit note** field, enter a Rule that stores the audit note that you want to include.
5. **Optional:** To indicate how you want to set properties for work items as they advance over this connector, in the **Set Properties** section, set the properties explicitly or with a Data Transform.

©2025 Pegasystems Inc

For more information, see Referencing properties, Data Transforms, and Setting property values with the Clipboard tool.

> **Result:**
>
> A Clipboard icon is displayed on the connector if you assign properties or a Data Transform on the connector.

6. Click **Submit**.
7. To define more connectors or alternate paths in the Process, repeat steps 2 through 6.

> **NOTE:** To prevent blocked Processes, always configure no more than one connector with the `Else` Condition type.

8. In the upper-right corner of the Flow form, click **Save**.

> **For example:**
>
> In the context of an application designed for selecting road assistance services, you can add an Assignment shape in the `Perform service` Process and select a Flow Action on the outbound connector associated with the newly added shape to facilitate confirmation of the request details:

©2025 Pegasystems Inc

Adding an Assignment shape to a Process

Result:

At run time, your application selects one connector to follow. You can test connectors by running your Process.

Related concepts

- Flow Actions
- Assignment shapes in Processes

Related tasks

- Adding a conditional path to an Assignment

Related reference

- Shape icons in a Process

©2025 Pegasystems Inc

Shape icons in a Process

Shape icons in a Process are a visual indicator of a shape's configuration and current processing. You can use shape icons to quickly review the functionality and progress of a Process without having to open the individual shapes. For example, you can view which Assignments have corresponding tickets, or if an Assignment generates any notifications.

You can view shape icons on Assignments when you edit your Process in a Process diagram. Shapes in a Process support the following icons:

Icon	Description
	The Assignment has one or more corresponding tickets. By creating a ticket, you can jump to a specific Assignment in a Process and skip a part of the processing. For example, a user can skip Assignments to provide the personal details of their spouse if they select an option that they are single in one of the previous Assignments. For more information, see Creating a ticket.
	The Assignment includes a Service-Level Agreement (SLA). For example, you can define a deadline of three business days to resolve an Assignment. For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.
	The Assignment generates one or more notifications. For example, you can notify

©2025 Pegasystems Inc

Icon	Description
	interested stakeholders about the progress of a Case. **NOTE:** You can add Notifications directly to Assignments only in Dev Studio. For more information, see Sending custom notifications about new Assignments.
	The Assignment includes routing configurations. For example, an application can route an Assignment to a current user or a Work Queue that a group of users can access. For more information, see Assigning tasks to users.
	A spin-off Subprocess runs simultaneously with the current Process. The current Process can end without completing the spin-off Subprocess first. For example, an office worker can work on a spin-off Subprocess of granting a new hire a parking spot in parallel to a main onboarding Process. For more information, see Adding additional configuration to a Subprocess.

©2025 Pegasystems Inc

Icon	Description
	When a Case reaches an Assignment, the Case Status changes to a value that includes the `New` prefix. For example, when a customer submits a new Case and the Case Status changes to `New-Ready`. For more information, see Changing Case Statuses.
	When a Case reaches an Assignment, the Case Status changes to a value that includes the `Open` prefix. For example, when an application routes a Case to a user for processing and the Case Status changes to `Open-InProgress`. For more information, see Changing Case Statuses.
	When a Case reaches an Assignment, the Case Status changes to a value that includes the `Pending` prefix. For example, when an application routes a Case to a manager for approval and the Case Status changes to `Pending-Approval`. For more information, see Changing Case Statuses.
	When a Case reaches an Assignment, the Case Status changes to a value that includes the `Resolved` prefix. For example, when

©2025 Pegasystems Inc

Icon	Description
	processing of a loan request ends and results in granting of the requested money, the Case Status changes to `Resolved-LoanGranted`. For more information, see Changing Case Statuses.

For example:

The following figure shows a sample Process of running a background check that includes Start and End shapes, three Assignments, and one Subprocess. The processing occurs in the following order:

1. The application routes the `Verify education` Assignment and the Case changes status to `Open-InProgress`.
2. A specific user processes the `Verify work experience` Assignment, which includes an SLA.
3. Simultaneously, a spin-off Subprocess named `Check social security` starts. Currently, the Subprocess does not include defined routing.
4. After the user completes the `Verify work experience` Assignment, the Case moves to the `Prepare for approval` Assignment and routes the Assignment according to a defined configuration. Additionally, the application sends a notification.
5. The End shape in the Process includes a ticket. If a use case does not require a background check process, an application can skip the Process and move straight to the End shape with the ticket.

©2025 Pegasystems Inc

Icons in a Running a background check process

- Process shapes
- Adjusting a Service-Level Agreement for the Assignment Service shape
- Responding to business exceptions in a workflow
- Calling one Process from another Process
- Changing Case Statuses

Types of Processes

The default method of creating Processes is by adding Steps when you define a Case Lifecycle. To access advanced options and configurations, you can also build your Processes by using Process shapes on a Process 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 Lifecycle. 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.

©2025 Pegasystems Inc

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 the 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.

©2025 Pegasystems Inc

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 Process 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

©2025 Pegasystems Inc

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 Dev Studio.

In Case Lifecycle Designer, a counterpart of a Screen Flow is a Multi-step Form. For more information, see Adding a Multi-step Form to a Stage.

Parent and Subprocess

A Process that calls another Process by using a relevant shape is the 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:

©2025 Pegasystems Inc

- 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 Running multiple instances of different Subprocesses with a Split Join shape.

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 the main Process and allows multiple users to complete work simultaneously, which saves time during Case resolution.

©2025 Pegasystems Inc

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 Lifecycle Designer, as shown in the following figure:

Parallel processing

For more information about creating parallel Processes in Case Lifecycle 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

©2025 Pegasystems Inc

Subprocess shape. For more information, see Adding additional configuration to a Subprocess.

For learning materials about parallel processing, see the Parallel processing in Pega applications module on Pega Academy.

- Creating a stand-alone Process

Assignment shapes in Processes

Assignment shapes represent tasks that users complete in a Process. To ensure that your business process meets all your unique requirements, you can add an Assignment shape for each place in a business process that requires human judgment and input.

For example, you can add Assignments to the Lifecycle of a Case to collect information from users with different roles or levels of expertise, such as a form that a recruiter completes after interviewing a job candidate and reviewing the list of necessary documents in the recruitment process. You can also increase the functionality of your application by assigning tasks to an external application or to a robotic queue.

Assignment shapes

You can create Assignments by adding certain Steps, such as the Collect information, Approve/Reject, or Form Steps, to a Process in Case Lifecycle Designer or by adding one of the Assignment shapes to a Process in Process Modeler, which gives you more configuration options.

You can add, and then configure, the following Assignment shapes:

Assignment

The basic Assignment shape that you use to collect data from users. The shape creates a task in a Work Queue or Worklist, so that a user can provide input to the Process. For example, you can use this shape in a mortgage Case Type to create a task for a Case worker to calculate a new interest rate.

©2025 Pegasystems Inc

For more information, see Process shapes and Collecting information from a user.

> **NOTE:** Some advanced settings for the Assignment shape are not available in Case Lifecycle Designer. You can access these settings on the Diagram tab of the Flow form by double-clicking an Assignment.

Assignment Service

An advanced shape that you use to continue your work flow in an external system. The shape calls an activity that requests and receives information from an external system by using a connect rule. For example, you can use the shape to retrieve a customer's financial history from an external system or department.

For more information, see Process shapes and Assigning a task to an external application.

> **NOTE:** Advanced shapes are available only in Dev Studio.

Assign to robot queue

An advanced shape that you use to call a robotic automation. For example, you can use this shape to run an Automation that analyzes the information submitted with an insurance claim.

For more information, see Process shapes and Assigning a task to a robotic Work Queue.

> **NOTE:** Advanced shapes are available only in Dev Studio.

©2025 Pegasystems Inc

Assignment settings

You can decide how users process the Assignments in your application by configuring the following settings:

Routing

Determines who processes the Assignment. In a typical scenario, a regular Assignment shape routes to a user (which in Dev Studio is known as the current or specific operator, or Worklist), or a team (which in Dev Studio is known as a Work Queue).

For more information about the routing options, see Collecting information from a user, Assigning users automatically at run time with business logic, and Configuring custom routing logic for Assignments.

The Service-Level Agreement

Defines the recommended and required completion times for the Assignment by creating goals and deadlines for Case workers.

For more information, see Completing work on time and Creating a Service-Level Agreement Rule in Dev Studio.

Assignment configuration

Defines Assignment details such as harness, confirmation note, and screen navigation.

For more information, see Harnesses, Sending confirmations of completed Assignments, and Creating navigation Rules.

Local Actions

Specifies local Flow Actions that users can choose to take before they complete the Assignment. At run time, local Actions are displayed in the header of a Case. To configure a local Action for an Assignment, you create or select an existing Flow Action. Submitting a local Action at run time does not complete the current Assignment.

PEGA ©2025 Pegasystems Inc

For more information, see Flow Actions.

Notifications

Sends out correspondence, such as an email message, when a running Process creates an Assignment. Typically, the application sends this correspondence to one or more of the work parties identified in the work item, and then reports progress to that party. Users can select a notify activity, which is an activity of the notify activity type, on the Security tab.

For more information, see Activity form - how to create activities for Processes.

Tickets

Determines whether an Assignment is associated with a ticket, which can be raised at any time by any Process in the same Case. Processing is connected to a ticket to respond to a business exception, error flow, or event, such as for example a user withdrawal from processing.

For more information, see Creating a ticket.

Optimization

Records work item properties that you want to use in the optimization analysis with the Process Optimization tool, which performs probability analysis on Assignments to identify common flow patterns and trends.

Assignment types for custom routing options

Each Assignment in a Process has a type based on the standard activity that creates the Assignment. You can select a standard activity when you define the routing options for an Assignment in a Process diagram in Case Lifecycle Designer. When you select the **Custom** routing option on an Assignment shape on the Diagram tab of the Flow form, you can then select an Assignment activity from the list in the **Assignment Type** field:

Connect

Creates an Assignment that waits for an external system to perform processing before moving to the next Step in the Process.

©2025 Pegasystems Inc

> **NOTE:** This Assignment type supports the Assignment Service shape, which is available only in Process diagrams. For more information, see Process shapes and Assigning a task to an external application.

Dependency

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

> **NOTE:**
>
> - This Assignment type is available only in Process diagrams.
> - As a best practice, use the standard Wait shape instead of a Dependency Assignment type. For more information, see Process shapes and Pausing and resuming Processes in Cases.
> - Use this Assignment type only if you want to fine-tune the waiting logic with a When Rule.

External

Uses the Directed Web Access feature to create an external Assignment, for example, to receive input from an external user.

For more information, see Directed Web Access in configuring Assignments for external users.

> **NOTE:** This Assignment type is available only in Process diagrams.

Work Queue

Creates an Assignment that an application can route to a Work Queue that is common to multiple users.

PEGA ©2025 Pegasystems Inc

For more information, see Configuring custom routing logic for Assignments, Creating a Work Queue and Assigning users automatically at run time with business logic.

Worklist

Creates an Assignment that an application can route to the Worklist of a current user or a specific user that the router activity defines.

For more information, see Configuring custom routing logic for Assignments and Assigning users automatically at run time with business logic.

- Collecting information from a user
- Step types
- Activity form - how to create activities for Processes
- Assigning tasks to users
- Flow Actions
- Automated and intelligent work assignment with Get Next Work
- Get Next Work feature module on Pega Academy

Annotating a Process

Promote sharing information with other members of your development team by annotating a Process. When you document the functionality of a Process, you can quickly communicate how a Process works or fits business requirements, and consequently speed up application development and avoid unnecessary changes that might come from insufficient understanding of a functionality. For example, by analyzing annotations, new team members can quickly understand why certain actions occur in a Process.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. On the **Workflow** tab, click **Life Cycle**.
3. Hover over a Process in a Stage, and then click the **Configure Process** icon.
4. On the toolbar, click **Flow Shapes**, and then click **Annotation**.

©2025 Pegasystems Inc

5. Double-click the Annotation shape, and then replace the default text with your notes or comments.
6. **Optional:** To adjust the size of the Annotation shape to fit the text, resize the Annotation shape.
7. Drag the Annotation shape to a different position on the canvas, based on the functionality that you want to document.
8. Click Save.

Result:

At run time, annotations have no effect on Processing.

For example:

The following figure shows a Process that includes an annotation. The annotation guides developers to maintain existing business logic criteria for routing an Assignment until the end of a release. As a result, the annotation minimizes the risk of unintentional changes that could impact business processes.

Annotation in a Process

- Changing the path of a Process
- Subprocesses

©2025 Pegasystems Inc

Keyboard shortcuts for Processes

For faster and more efficient creation of Processes, you can use keyboard shortcuts in the Process diagram. Using keyboard shortcuts also supports accessibility and provides a better experience for developers who rely on a keyboard while developing an application.

You can use the following keyboard shortcuts when you create a Process in a Process diagram:

Shortcut	Description
Shift + mouse click	Selects an item or removes it from the rest of the selected items. For example, you can select multiple Process shapes so that you can delete more than one shape at once.
Delete or Backspace	Deletes the selected items.
Ctrl + A	Selects all objects on the canvas.
Alt	Temporarily switches between pan mode and select mode.
Secondary mouse button + drag	Temporarily pans the canvas when in select mode or temporarily selects items when in pan mode.
+ (Windows) or = (Mac OS)	Zooms in.
-	Zooms out.
Ctrl + M	Toggles between full screen and the normal canvas.
Esc	Exits the current operation.
Ctrl + S	Saves changes.

PEGA ©2025 Pegasystems Inc

Shortcut	Description
Double click on the canvas	Centers and fits the screen diagram.

- Configuring an accessible UI

Creating a Screen Flow in Dev Studio

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

For example, in an application to book medical appointments, you can create separate screens for patients to enter their personal details, their medical history, and a description of their current issue. As a result, the patient interacts with logically organized information.

> **NOTE:** Creating a Screen Flow in Dev Studio provides for advanced configurations and is suitable for advanced developers. To create a form with multiple screens by using low-code tools, switch to App Studio and create a Multi-step Form. For more information, see Adding a Multi-step Form to a Stage.

Screen Flow is a type of a Process. For more information, see Types of Processes.

1. Create the Rule that stores the Screen Flow:
   a. In the header of Dev Studio, click **Create menu** > **Process** > **Flow**.
   b. In the **Label** field, describe briefly the purpose of the Screen Flow.

      > **For example:**
      >
      > Enter `Collect appointment booking info`.

©2025 Pegasystems Inc

c. **Optional:** To change the default identifier for the Screen Flow, click **Edit**, and then provide a unique value in the **Identifier** field.
d. Click **View additional configuration options**, and then select **Standard template for Screen Flows**.
e. In the **Context** section, select an application layer to store the Screen Flow.
f. In the **Apply to** field, press the Down arrow key, and then select the class that defines the scope of the Screen Flow.
   The class controls which Rules the Screen Flow can use, as well as which Rules can call the Screen Flow.
g. In the **Add to Ruleset** field, select the name and version of the Ruleset that stores the Screen Flow.
h. Click **Create and open**.

Result:

A Process 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:

Screen Flow

2. Open a dialog box with Screen Flow configuration by double-clicking the Start shape.

©2025 Pegasystems Inc

3. **Optional:** To provide a unique name for the Start shape, in the **Start Shape** field, enter a new value.
4. In the **Start** section, define how an application displays the Screen Flow:

Choices	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	a. In the **Harness** list, select **Custom**. b. 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: *Tabs in a Screen Flow*
Display navigation to screens as a tree view	a. In the **Harness** list, select **Custom**. b. 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

©2025 Pegasystems Inc

Choices	Actions
	layout, as shown in the following figure: *Tree layout of a Screen Flow*
Apply your custom harness	a. In the **Harness** list, select **Custom**. b. In the **Harness name** field, press the Down arrow key, and select your custom harness.

> **NOTE:** At run time, an application displays every Assignment in a Screen Flow as a separate screen.

5. **Optional:** To save a Case after the last Step in a Screen Flow is complete, select the **Save on last Step** checkbox.
   If you leave the **Save on last Step** checkbox clear, the Case saves information each time a user completes an Assignment. As a best practice, save the Case data on the last Step to avoid performance issues.
6. **Optional:** To allow users to move to the next Assignments in a Screen Flow even if an Assignment fails validation, select the **Allow errors** checkbox.

> **NOTE:** An option to allow errors is available only after you configure the Screen Flow to save a Case on the last Step.

©2025 Pegasystems Inc

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 Process passes validation on the last Step.

7. In the **Status** section, in the **Work status** field, press the Down arrow key, and then select a status that a Case obtains when the Screen Flow starts.

For example:

Select **Open-InProgress**.

8. In the **Routing** section, in the **Route to** list, determine a user that completes the Screen Flow:
   - To assign the Screen Flow to a user who currently processes the Case, select **Current operator**.
   - To assign the Screen Flow to another user in your application, select **Operator**, and then configure the user parameters.

     You can assign the Screen Flow to a user by user name or user reference, the reporting manager of the user who currently processes the Case, or a Case participant.

   - To assign the Screen Flow to a team that shares a Work Queue, select **Work Queue**, and then select a team.

     A user who picks up the Assignment from a Work Queue needs to complete an entire Screen Flow.

   - To assign the Screen Flow to a Worklist or a team that shares a Work Queue, select **Custom**, and then in the **Router** field, enter a router activity to route the Assignment.

     For example, you can route a Screen Flow to an external Work Queue.

©2025 Pegasystems Inc

- 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 Process shapes in the Screen Flow.

9. **Optional:** To define business requirements and instructions for the Process, in the **Specification** section, associate a specification with your Process:
   - To use an existing specification, in the **Specification** field, enter the specification.
   - To create a new specification, enter text in the rich text editor.
   - To use a `.docx` file that stores a specification, click **Upload Word document as description**, and then in the dialog box, select the file that you want to use.
10. Click **Submit**.
11. Click **Save**.

What to do next:

Expand your Screen Flow with more Process shapes, and then integrate the Screen Flow with your Case Lifecycle:

1. 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 Process shapes.
2. Mark the Screen Flow as a relevant record, so that you can reuse the process across your application. For more information, see Marking a record as relevant.

©2025 Pegasystems Inc

3. Add the Screen Flow to your Case Lifecycle. For more information, see Adding a sequential Process to a Stage.

- Collecting information from a user
- Assigning tasks to users
- Decision trees

Changing the path of a Process

You can change the path of a Process to support out-of-sequence events. By adding different structures to your Process, such as loops and branches, you can improve the flexibility of Cases that you create.

The following Tasks can help you change the path of a Process:

- **Adding Decisions to Processes**
- **Adding a conditional path to a Process**
- **Adding a conditional path to an Assignment**
- **Adding a results-based path to 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.

For relevant training materials, see the Automating workflow decisions module on Pega Academy.

©2025 Pegasystems Inc

Before you begin:

Configure a Process:

1. Add a Process to a Stage in your Case Lifecycle. For more information, see Processes in a Case Lifecycle.
2. Define the Steps for your Process. For more information, see Adding Steps to Processes.

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 **Lifecycle**.
3. In the **Case Lifecycle** 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.

©2025 Pegasystems Inc

8. Click the Decision shape.
9. In the Decision shape property pane, 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.

©2025 Pegasystems Inc

Choices	Actions
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. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
13. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

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**
- **Changing the decision logic in a Process**
- Conditionally starting a Process
- Troubleshooting connector configuration issues for Cases in flight

PEGA ©2025 Pegasystems Inc

Types of decision logic

You can use different types of logic when you add the Decision shape to a Process. By understanding what makes each type unique, you can create Processes that make decisions more effectively.

You can use the following types of decision logic in a Process:

- **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 Process

Changing the decision logic in a Process

You can change the way that you implement a decision in a Process. By using a Rule instead of a series of When Conditions to define your decision logic, you can create Cases that support complex decisions and are easier to maintain.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Workflow** tab, click **Lifecycle**.
3. Click a Process that contains a Decision Step, and then click **Open Process**.

PEGA ©2025 Pegasystems Inc

4. On the **Diagram** tab of the Flow form, double-click the Decision shape to open the property pane.
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:

     - In the **Rule** field, press the Down Arrow key and select the Rule that defines your decision logic.

     - Provide input values, based on the type of Rule that you select.

       - **Input** - Passes text or a property reference to a decision tree.

       - **Row input** - Passes text or a property reference to a row in a map value rule.

       - **Column input** - Passes text or a property reference to a column in a two-dimensional map value rule.

     - In the **Result** in field, enter the name of a property that stores the result of the decision.

   - For decision logic that you define in a Boolean expression:

     - In the **Expression** field, enter a Boolean expression that defines your decision logic.

     - To access helpful tips and built-in functions for your expression, click the **Build an Expression** icon.

   - For decision in a fork. Percentage — Follows the connector for a specific percentage of flow runs.

PEGA ©2025 Pegasystems Inc

For example, you can ensure that 20 percent of mortgage applications take a different path in the process 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 pane.
8. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
9. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Adding Decisions to Processes
- Decision tables
- Decision trees
- Scorecard Rules
- Predictive Model Rules
- Map values
- Building Expressions with the Expression Builder

Adding a conditional path to a Process

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

1. Open a Process by searching for it or by using the Application Explorer. For more information, see Finding Rules by class.

©2025 Pegasystems Inc

2. Define a new path by connecting a sequence of shapes to an existing shape in the Process. For more information about the different shapes that you can use, see Flow shapes.
3. Configure the Conditions that cause the Process to follow the new path.
   a. Double-click the connector that goes from the existing shape in the Process to the first shape in the new path.
   b. In the **Condition type** list, select **When**.
   c. In the **When** field, enter a When Condition that your application evaluates at run time.
   d. In the **Likelihood** field, enter a number between 1 and 100 to control the order in which your application evaluates the connector at run time.
   e. Click **Submit**.
4. Review the configuration of all other paths, or connectors, from the shape to ensure that it supports the following best practices:
   - There is only one **Else** connector, so that the flow can continue when no Conditions return a true value.
   - There are no **Always** connectors, which are applicable only when there is a single path from one shape to another.
   - Each connector provides a unique *Likelihood* value.
5. Click **Save**.

- Changing the path of a Process
- Adding a results-based path to a Process
- 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 Process 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.

©2025 Pegasystems Inc

1. Open a Process 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 Process.
   b. In the **Likelihood** field, enter an integer between 1 and 100 to control which connector the Process 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 Process 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 Process 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** checkbox.
   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.

PEGA ©2025 Pegasystems Inc

   e. In the **When** field, press the Down Arrow key and select a When Condition that controls whether the Process follows the new connector. At run time, When Conditions are evaluated based on the list order that you provide.
   f. **Optional:** To update Case information after the Flow Action is complete, open the Flow Action and define a post-processing activity or Data Transform.
   For more information, see Configuring post-processing for a Flow Action
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 Process

You can use a connector to add a results-based path to a Process. By using run-time values to determine which path a Process 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 Process 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 Process. For more information about the different shapes that you can use, see Flow shapes.
3. Configure the result, or return value, that causes the Process to follow the new path.

©2025 Pegasystems Inc

a. Double-click the connector that goes from the existing shape in the Process to the first shape in the new path.
b. In the **Condition type** list, select **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` 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 Process 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 in Dev Studio
- Creating Views for Case Types

Subprocesses

A Subprocess is a Process that starts after another Process calls it. To create a Subprocess, add a specific Step to a Process in the Lifecycle of a Case or add specific shapes to a Process diagram.

PEGA ©2025 Pegasystems Inc

Subprocesses modularize the events in the Lifecycle of a Case and increase Reusability of your resources, as you can implement the same Subprocess in multiple scenarios.

In a sample scenario, you can run a Subprocess of collecting user feedback in multiple Case Types, such as reviewing a loan request, resolving a credit card dispute, and processing an insurance claim. Additionally, Subprocesses can reduce Case resolution time because they allow for processing of Case events in parallel. For example, in a Case of preparing onboarding for a new hire, you can create Subprocesses for an IT setup and facilities setup. Different workers can move each Process forward, independently and in parallel, which leads to faster Case resolution.

You can configure different types of Subprocesses to meet your business requirements. When implementing Subprocesses into your Case Types, you have the following options:

Process Step

In Case Lifecycle Designer, you can add a Step that represents an existing Process in your application. A Case moves forward after the Subprocess finishes. For example, in a Review Stage of a Hire a job candidate Case Type, you can add a Subprocess that represents running a background check. Internally, the Subprocess consists of a series of Steps, such as checking the criminal record and social security. As a result, you can maintain all the common business functionality with the Subprocess for easier updates and clarity. Also, adding an entire Process instead of single Steps saves time and promotes the reuse of your application resources.

The following figure shows a Process Step in a Review Stage:

©2025 Pegasystems Inc

Process Step

For more information, see Calling one Process from another Process.

Subprocess shape

In a Process diagram, you can add a Subprocess shape to a Process in your Case Type. In the background, a Subprocess shape is the same entity as a Process Step that you add through Case Designer, and both options have the same run-time effect. However, by adding a Subprocess shape through a Process diagram, you have more configuration options than a Process Step. Additionally, in Dev Studio, you can configure advanced options for the Subprocess shape, such as defining a working context for a Case. You can also start a Subprocess as a spin-off. As a result, the Subprocess runs independently from the main Case, and the Case can move forward without waiting for the Subprocess to finish. For example, in an Items purchase Case Type, if the customer wants to buy more items than are currently in stock, an application can start a spin-off Subprocess to order more items from a supplier. The

©2025 Pegasystems Inc

Subprocess runs independently, and the Items purchase Case can reach resolution without waiting for the Subprocess to rejoin.

The following figure shows a Process diagram with a Process that includes an `Order from a supplier` Subprocess. The arrows icon indicates that the Subprocess is a spin-off. The image includes also a dialog box with Subprocess configuration.

Spin-off Subprocess

For more information, see Adding additional configuration to a Subprocess.

©2025 Pegasystems Inc

Split Join shape

In a Process diagram in Dev Studio, you can add a Split Join shape

that starts multiple instances of different Subprocesses that later rejoin the main Process. Based on the configuration that you provide, a Case moves forward when one or more Subprocesses rejoin the main Process. For example, in a Case of reviewing a loan request, you can use a Split Join shape to start Subprocesses of reviewing a monthly income and banking history of an applicant. Different Case workers can work on those Subprocesses in parallel, which saves time.

The following image shows a Process diagram with a Process that includes the Split Join shape. The Split Join shape calls two Subprocesses: Check monthly income and Check banking history.

Split Join shape in a Process

For more information, see Running multiple instances of different Subprocesses with a Split Join shape.

A Split for Each shape

In a Process diagram in Dev Studio, you can add a Split for Each shape

that runs multiple instances of the same Subprocess. Based on the configuration that you provide, a main Process resumes after one or more Subprocesses rejoin the main

©2025 Pegasystems Inc

Process. You can also iterate a Subprocess through all items on a page list that you provide, such as a list of managers. For example, in a Hire a new candidate Case Type, you can use a Split for Each shape to route a Review documents Subprocess to multiple managers that need to review documents from a job candidate. The Split for Each shape routes work dynamically by using a page list, so in one instance of the Hire a new candidate Case three managers might need to complete the Review documents, while in another instance the approval is required only from two managers, as the Split for Each shape generates the Review documents Subprocess for each manager added to the Case. Configuration options determine if all managers, any, or some need to approve before the Case can proceed. The managers can review documents and complete the Subprocess in parallel.

The following figure shows a Process diagram with a Process that includes the Split for Each shape and a dialog box with a shape configuration. The Split for Each shape routes a Review documents Subprocess to all managers that are listed in the *.ListManagers* page list. The system populates the page list dynamically for each instance of a Case at run time.

©2025 Pegasystems Inc

Split for Each shape in a Process

For more information, see Running multiple instances of the same Subprocess with a Split for Each shape.

- **Calling one Process from another Process**
- **Adding additional configuration to a Subprocess**
- **Running multiple instances of different Subprocesses with a Split Join shape**

©2025 Pegasystems Inc

- **Running multiple instances of the same Subprocess with a Split for Each shape**

- Processes in a Case Lifecycle
- Adding a sequential Process to a Stage
- Adding a parallel Process to a Stage
- Creating a stand-alone Process

Calling one Process from another Process

Make your business processes more granular by complementing a Process with a Subprocess in your Case Lifecycle. As a result, you avoid creating long and complex Case types and deliver an application that is easier to maintain and edit if your business requirements change. Additionally, you save time because you reuse assets across your application.

For example, in a loan request Case Type, you can create a Process that represents revising a banking history of a customer, and then complement it with a Subprocess of revising monthly regular payments. You can also reuse the Subprocess for other Case Types, such as a mortgage request.

> **Before you begin:**
>
> Ensure that the subprocess that you want to add already exists in your application. For more information, see Processes in a Case Lifecycle.

When you add a Subprocess to your Case Lifecycle, you add a whole set of Steps at once. Consequently, you save even more time, because you avoid adding and configuring individual Steps.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, locate the Process that you want to complement with a Subprocess, and then click **Step** > **More** > **Processes**.

©2025 Pegasystems Inc

> **TIP:** You can adjust the list of Subprocesses by managing relevant records in your application. For more information, see Adding a relevant record to a specified class in your application.

3. **Optional:** To adjust the Subprocess name to your current business needs, in the Subprocess Step, in the field, enter a new name.
4. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **What to do next:**
>
> 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. For more information, see Adding additional configuration to a Subprocess.

- Subprocesses
- Adding additional configuration to a Subprocess
- Creating a stand-alone Process

Adding additional configuration to a Subprocess

Provide access to relevant resources, such as properties or Rules, that are outside your current Case Type by changing the scope of a Subprocess in the Case Type. As a result, you ensure that users can access information that is necessary to successfully resolve their business processes even if the resolution requires data from another Case.

©2025 Pegasystems Inc

For example, a customer service representative (CSR) can resolve a complaint about hotel service by using information that a customer provided while booking a hotel room.

> **Before you begin:**
>
> Add a Subprocess to your Case Lifecycle. For more information, see Calling one Process from another Process.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, in the header of a Process that contains a Subprocess, click **Configure Process**.
3. On the toolbar, click **Open Process**.
4. On the **Diagram** tab of the Flow form, open the property dialog box by double-clicking the Subprocess shape.
5. In the **Subprocess details** section, in the **Define flow** field, select a working context for the Subprocess:

Choices	Actions
The Subprocess works on a current Case	Select **On current page**.
The Subprocess works on a specific Case	a. Select **On specific work item**. b. In the **Work property** field, enter a property reference in the current work item that identifies the key of the Subprocess work item. c. In the **class** field, enter the class of the Subprocess work item. d. **Optional:** To specify a Clipboard page that holds the Subprocess

©2025 Pegasystems Inc

Choices	Actions
	work item, in the **Page name**, enter the Clipboard page name. **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.
The Subprocess works on an embedded page	a. Select **On embedded page**. b. In the **Page property** field, enter a reference to the property of the Page, Page list property, or Page group property type that holds the embedded page. c. In the **class** field, enter the class of the embedded page.

6. In the **Flow name** field, enter a name of the Process that includes the Subprocess for which you provide additional configurations.

©2025 Pegasystems Inc

By default, the system autopopulates this field with the name of the current Process.

7. **Optional:** To display a message in the audit trail of the Subprocess, in the **Audit note** field, enter a Rule that stores the audit note that you want to include.
8. **Optional:** To allow the parent Process to continue processing without waiting for this Subprocess to return, select the **Spinoff flow** checkbox.

   Result:

   When a Case reaches the Subprocess that runs as a spin-off, the Subprocess runs asynchronously and independently from the main Process. The main Process can continue without waiting for the results from the Subprocess. The main Process and the Subprocess might operate on the same work item or different work items.

9. **Optional:** To provide a link to the Subprocess in the breadcrumb trail navigation, select the **Enable navigation link** checkbox, 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** checkbox.
   - 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** checkbox.
   - To allow users to return to the last Step in the Subprocess, select the **Subprocess has navigation links** checkbox.

   NOTE: If you select neither the **Only allow navigating back to this Step** nor **Subprocess has navigation links**, the users can go back only to the first Step in the Subprocess.

10. Click **Submit**.
11. In the upper-right corner of the Case Type, save your changes by clicking **Save**.

©2025 Pegasystems Inc

12. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Calling one Process from another Process
- Subprocesses

Running multiple instances of different Subprocesses with a Split Join shape

Divide your independent business requirements into multiple Processes by calling two or more Subprocesses that later rejoin the parent Process. As a result, you develop a granular and flexible application that you can conveniently adjust to your changing business requirements instead of creating long and complicated Case Types.

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

> ⓘ **NOTE:** Each Subprocess runs asynchronously and in parallel. When a user works on an Assignment in one Subprocess, other users can work on an Assignment from a different Subprocess.

1. Add the **Split Join** shape to your Process:
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. In the **Case Lifecycle** section, in the Process in which you want to add the Split Join shape, click **Configure Process**.

©2025 Pegasystems Inc

   c. On the toolbar, click **Open Process**.
   d. On the **Diagram** tab, click the **Flow Shapes** icon, and then select **Advanced Shapes** > **Split Join**.
   e. Connect the **Split Join** shape by dragging connector end points to connection points on different shapes in the Process.
      Connectors that work correctly are green.
2. Open the dialog box by double-clicking the **Split Join** shape.
3. **Optional:** To provide a unique name for the shape, in the **Split Join** dialog box, in the **Split Join** field, enter a new name.
4. In the **Join** list, define when the parent Process resumes processing:

Choices	Actions
Resume processing after all Subprocesses are complete	Select **All**.
Resume processing after any of the Subprocesses is complete	Select **Any**. After any of the Subprocesses are complete, processing of the remaining Subprocesses stops and the system cancels open Assignments.
Resume processing after a When Condition returns a true value	a. Select **Some**. b. In the **Exit iteration** list, select **On when**. c. In the **When** field, enter a When Condition that resumes processing after evaluating to true.
Resume processing after a specified number of Subprocesses reach a certain status	a. Select **Some**. b. In the **Exit iteration** list, select **On count**.

©2025 Pegasystems Inc

Choices	Actions
	c. 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 Subprocesses in your Process. d. In the **Resulting with flow status** field, enter a status that Subprocesses need to reach to resume processing. **For example:** Enter `Resolved-Completed`.

5. **Optional:** To provide a link to the Split Join shape Step in the breadcrumb trail navigation, select the **Enable navigation link** checkbox, and then configure additional navigation options:
   - To enable users to return to the Step after the Case moves forward, select the **Only allow navigating back to this Step** checkbox.
   - To enable post-processing or validation when users navigate back from the Split Join shape Step by using the breadcrumb trail, select the **Perform post-processing when navigating away from Step** checkbox.

©2025 Pegasystems Inc

6. Open the Subprocess configuration by expanding the **Specify a flow Rule for this Subprocess** section.
7. In the **Name** field, enter a clear description of the Subprocess.
8. On the **Subprocess** tab, configure the context of the Subprocess:

Choices	Actions
The Subprocess works on a current Case	Select **On current page**.
The Subprocess works on a specific Case	a. Select **On specific work item**. b. In the **Work property** field, enter a property reference in the current work item that identifies the key of the Subprocess work item. c. In the **class** field, enter the class of the Subprocess work item. d. **Optional:** To specify a Clipboard page that holds the Subprocess work item, in the **Page name**, enter the Clipboard page name. **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

©2025 Pegasystems Inc

Choices	Actions
	`pyNextObj`, `pyNextObj_1`, and so on.
The Subprocess works on an embedded page	a. Select **On embedded page**. b. In the **Page property** field, enter a reference to the property of the Page, Page list property, or Page group property type that holds the embedded page. c. In the **class** field, enter the class of the embedded page.

9. In the **Filter by flow** field, select the Subprocess category:
   - To start a Subprocess that consists of a series of actions to complete, select **Process flow**.
   - To start a Subprocess that consists of a series of forms to complete, select **Screen Flow**.
10. In the **Flow Rule** field, press the Down arrow key, and then select the name of a Subprocess to run.
11. **Optional:** Specify what information about the Subprocess you want to include in the application documentation:
    - To link an application to the Subprocess, in the **Application** field, enter the name of the application.
    - To provide a diagram of the Subprocess in a documentation section that describes a specific Case Type, in the **Case/Supporting type** field, enter the name of the Case Type.
    - To provide implementation requirements of the Subprocess, in the **Specification** field, enter the specification that holds the requirements.

©2025 Pegasystems Inc

12. **Optional:** To display a message in the audit trial of the Subprocess, in the **Audit note** field, enter a Rule that stores the audit note that you want to include.
13. Configure the second Subprocess by repeating steps 6 through 10.
14. **Optional:** To add more Subprocesses, click **Add a flow**, and then repeat steps 6 through 10.
15. Click **Submit**.
16. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
17. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Creating a stand-alone Process
- Changing the path of a Process
- Subprocesses
- Process shapes

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.

PEGA ©2025 Pegasystems Inc

> **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 Lifecycle:
   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 Lifecycle** section, in the Process in which you want to add the Split for Each 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**.

©2025 Pegasystems Inc

- To resume processing after a specified number of Subprocesses is complete, select **Some**.

5. In the **Page property** field, enter the Page list property or Page group property 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 selected **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 selected **Some** and your Process resumes after a specified number of Subprocesses is complete, configure the Conditions to resume processing:

©2025 Pegasystems Inc

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`.

©2025 Pegasystems Inc

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** checkbox, 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** checkbox.
    - 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** checkbox.
    - 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** checkbox.
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** checkbox.

    > **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** checkbox. Otherwise, your Process evaluates the pages without a match after processing all Conditions in the **Page group iteration settings** section.
14. Click **Submit**.

©2025 Pegasystems Inc

15. In the upper-right corner of the Case Type, save your changes by clicking **Save**.
16. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Changing the path of a Process
- Process shapes

Flow Actions

Flow Actions are used to define what a user can do at a specific Step in a Case.

Flow Actions are Rules that are designed to control how users interact with forms to complete Assignments. At run time, Flow Actions determine the choices available to users as they perform an Assignment. For example, in a damaged goods claim process, the buyer can choose to have the goods either reimbursed or replaced. These Flow Actions might require different input fields and may use different form displays.

You use Flow Actions to define the user interactions that occur at each Step of the Process. For example, you can use Flow Actions to create the following user interactions:

- When a user starts the account opening process, you use a Flow Action to display a form with fields, buttons, and other UI elements that enable the user to enter and submit their information.
- When a customer service representative (CSR) reviews the form, you use a Flow Action to display a form that shows the information submitted by the user, so that the CSR at the bank can review and verify it. You can also use a Flow Action to define the actions that the bank can take, such as approving or rejecting the form, or requesting more information from the user.

PEGA ©2025 Pegasystems Inc

What are Flow Actions

Flow Action Rules are part of the Process category. More specifically, a Flow Action is an instance of the *Rule-Obj-FlowAction* Rule Type. In the navigation pane of Dev Studio, you use the Records Explorer to list all the Flow Actions available to you by clicking **Records** > **Process** > **Flow Actions**.

Whenever you add a Collect information Step or a Multi-step Form to a Case Lifecycle, the system creates a Flow Action for these Steps in the background, but you can also add and configure your own Flow Actions that are specific to your business needs. For more information, see Creating Flow Actions.

Where Flow Actions are referenced

In a Process, you can associate a Flow Action with the connector or Assignment shapes. For more information, see Configuring indicator for a Flow Action.

Flow Actions types

Based on the association with the Process element and role in advancing Case processing, Flow Actions can have the following types:

Connector Actions

- Completes the Assignment and advances the Case to the next shape.
- Configured in Process Modeler as a line shape, a connector exits from an Assignment or Assignment Service shape.

©2025 Pegasystems Inc

Connector Actions in a Process

- Can specify values for the likelihood parameter that indicate the expected percentage of times users are likely to choose each Action.

For more information, see Connecting shapes in a Process and Adding a conditional path to an Assignment.

Local Actions

- Updates the Assignment but does not advance the Case to the next shape; the Assignment is still open in the user's Worklist.
- Configured in the properties pane of an Assignment shape in Process Modeler.

Local Action configured in an Assignment shape

©2025 Pegasystems Inc

- Serve as an alternative to autogenerated controls. For more information, see Adding UI components to a View.
- Can be added as optional Actions to a Case Type. Flow Actions used as optional Actions in a Case Type are typically assigned to the Case rather than a Stage so that users, such as Case workers or managers, can perform these Actions at various points in the Case Lifecycle. For more information, see Configuring and working with optional Actions in Cases and Enabling bulk processing of Cases in Constellation.

Flow Actions at run-time

At run time, the system presents available Flow Actions in a selection list, with the Connector Flow Actions, sorted by decreasing likelihood, appearing above the Local Flow Actions.

> **NOTE:** In general, you cannot control the order of the Local Flow Actions. However, if for one Assignment you want to make a Local Flow Action appear first, add a connector to the Process that loops back to the Assignment, and assign the largest likelihood value to that connector.

Define your Flow Actions by completing the following tasks:

- **Creating Flow Actions**
- **Defining the run-time presentation of Flow Actions**
- **Configuring pre-processing for a Flow Action**
- **Configuring post-processing for a Flow Action**
- **Configuring back-to-back processing for a Flow Action**
- **Configuring Action buttons for a Flow Action**

©2025 Pegasystems Inc

- **Configuring indicator for a Flow Action**
- **Aligning Views to Flow Actions through naming**
- **Adding optional Actions to paused Processes**
- Adding a Validation Rule to a Flow Action
- Migrating Flow Actions
- Subprocesses
- Calling one Process from another Process
- Assigning tasks to users
- Assignment shapes in Processes
- Running Flow Actions in bulk
- Adding a Validation Rule to a Flow Action
- Viewing authorizations for Processes and Flow Actions in a single Access Group
- Step types

Creating Flow Actions

Define user interactions with forms in a Case by creating Flow Actions that you associate with Assignments, which represent tasks that users or groups receive.

With Flow Actions, users can provide data that is relevant to a particular Step in the Process. This data can help make decisions, trigger Actions, or update Case information. For example, in the process of resolving a customer inquiry, the `Review and Response` Flow Action serves as a key interaction point for a customer service representative (CSR). As a result, the CSRs can input a response and select the priority level, and the Flow Action triggers further Actions based on the selected priority.

1. In the header of Dev Studio, click **Create menu** > **Process** > **Flow Action**.
2. On the **Create Flow Action** form, enter values in the fields to define the context of the Flow Action:
   a. In the **Label** field, enter text that describes the purpose of the Action.

©2025 Pegasystems Inc

> **NOTE:** Make the description meaningful in context to application users who must select one Action from the list. As a best practice, use an action verb. For example, the following Action labels convey the intent of users clearly:
> - `Submit PO for approval`
> - `Perform quality review`
> - `Verify application`

b. **Optional:** To change the default identifier for the circumstance definition, click **Edit**, and then in the **Identifier** field, provide a unique value.
c. **Optional:** To include all other configuration options supported by this record type on the form, click **View additional configuration options**.
d. In the **Context** section, specify where the record resides in your application Ruleset stack and its resusablility in the class hierarchy.
e. In the **Apply to** field, select a class.
f. In the **Add to Ruleset** field, select the name and version of a Ruleset that stores the Flow Action.

3. **Optional:** To override the default work item that your application associates with this development change, in the **Current work item** section, in the **Work item to associate** field, select a work item.
   For more information about your default work item, see Setting your current work item.
4. Click **Create and open**.

Related concepts

- Restricting who can use a Flow Action

Related tasks

- Adding a Validation Rule to a Flow Action

©2025 Pegasystems Inc

- Controlling access to Flow Actions
- Viewing authorizations for Processes and Flow Actions in a single Access Group

Defining the run-time presentation of Flow Actions

Define a View or Section that the system displays in a form that opens when a user selects a Flow Action. Defining a View or Section that displays in a form upon selecting a Flow Action streamlines user interactions by presenting relevant information and input fields contextually, improving task efficiency and reducing processing time.

You design the form layout by defining a View reference or Section reference on the **Layout** tab of the Flow Action Rule. Flow Action Rules support both View and Section references, which helps you use the same Action in both traditional UI and Constellation UI applications. Constellation applications support View references. For more information, see Choosing a UI. Traditional UI applications support Section references.

> **NOTE:** You cannot use the **Layout** tab if you selected the **Reference HTML** checkbox on the **HTML** tab of the Flow Action Rule.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. Search for and open the Flow Action that you want to configure.
4. Depending on the UI that your application uses, define the display of the Flow Action:

Choices	Actions
Reference a View (Constellation applications)	In the **View configuration** section, in the **View** field, select a View.
Reference a Section (traditional UI applications)	a. In the **Section configuration** section, in the **Page context** field,

©2025 Pegasystems Inc

Choices	Actions
	select or confirm the page context for the Section. b. If in the **Page context** field you select `data page`, `clipboard page`, or `page defined by property`, in the **Page** field, provide the name of the page that is the basis for fields within the Section. Leave this field blank if the page is the one that matches the *Applies To* key part of the Rule. c. In the **Applies to** field, select the class of the Section that you want to reference. d. In the **Section** field, select the Section that you want to display in a form. e. **Optional:** To configure the Section that you selected, click the **Target** icon, and then, on the **Edit Section** form, modify the Section. For more information, see Working with Dev Studio forms. f. **Optional:** To enter parameters for the Section, click **Edit parameters**, and then define the parameters. **For example:** You can provide parameters automatically by selecting

©2025 Pegasystems Inc

Choices	Actions
	the Pass current parameter page checkbox.

5. **Optional:** To configure a dynamic behavior in a form field, specify the properties to monitor for changes and a mechanism that triggers actions when the properties change:

> **NOTE:** Form refresh conditions are executed on the server side. If a change in the form triggers an error message based on the refresh condition, the message does not prevent the user from submitting incorrect data. To prevent the user from submitting incorrect data, add a Validation Rule to your Flow Action. For more information, see Adding a Validation Rule to a Flow Action.

Choices	Actions
Constellation applications	a. In the **Form refresh settings** section, click Add Condition. b. In the When Property changes field, specify a property to monitor for changes. **NOTE:** Only the property change event is supported. Only Scalar properties (also known as Single-value property) on single Embedded pages

©2025 Pegasystems Inc

Choices	Actions
	(including the Case Reference, Data Reference, and Query field types) are supported. The Page list property type is not supported. For more information, see Properties. c. In the **then run Data Transform** field, specify a Data Transform to run when the property changes. This mechanism ensures that the form is updated with the default values. **NOTE:** If the logic to be run is only associated with the property that is changing, then use the *On Change - Refresh* Data Transform. If the logic is generic to any refresh, then use the *pyRefreshData* Data Transform. When running a Data Transform for properties of embedded pages, the system runs it in the Case

©2025 Pegasystems Inc

Choices	Actions
	context, not in the context of the embedded page. For example, if you set *Address.pyCity* as the watch, the system runs the Data Transform in the Case context, not in the *Address* page context. d. To add more Conditions, repeat steps 5.a through 5.c. **For example:** In this sample configuration of the refresh settings for a Flow Action, when the *UseBillingAddress* property changes, the system runs a Data Transform to copy the billing address to the shipping address: *Refresh settings for a form* For more information, see Configuring dynamic field behavior.

©2025 Pegasystems Inc

Choices	Actions
Traditional UI applications	In the **Refresh settings** section, in the **Refresh condition** field, enter a When Condition to trigger a refresh based on a property change or the addition or deletion of a row or column in a repeating layout. For more information, see When Condition Rules and Layouts.

6. **Optional:** To restrict the presentation of the View or Section in the read-write mode to those users who hold a specific privilege, in the **Security** section, in the **Update privilege** field, select a privilege name.

> **TIP:** You can also click the **Target** icon to review or create a privilege Rule. For more information, see Role permissions and privileges in Access Manager and Requiring a privilege for a Rule.

7. Save the Flow Action Rule.

What to do next:

To make the Flow Action available for reuse in App Studio, for example, while reusing the Flow as a smart shape, mark the Flow Action as a relevant record. For more information, see Marking a record as relevant.

Processes can be reused in App Studio as smart shapes, encapsulating advanced and complex functionalities in a low-code element that developers and business representatives can incorporate into their Case Types in App Studio. For more

©2025 Pegasystems Inc

information about Studios interoperability and modular reuse, see Reusable elements in App Studio.

Related tasks

- Accessing Case Views
- Creating Views
- Migrating Flow Actions

Related information

- Pega Academy Mission: Pega Constellation View-Based UI Configuration for Low-Code Authors
- Pega Academy Mission: Pega Constellation View-Based UI for Designers

Configuring pre-processing for a Flow Action

Provide relevant data before users start working on an Assignment by configuring pre-processing for a Flow Action. Flow Actions can call a Data Transform, an activity, and a robotic Automation, which gives you a possibility to meet your unique business requirements connected to data manipulation in a Case. For example, you can autopopulate fields on a form that a user needs to process to complete the Assignment.

In pre-processing, the system first applies a Data Transform, runs an activity, and then runs an Automation. Specify the elements of pre-processing that match your needs. For example, you can specify only a Data Transform, and you can omit providing robotic Automation and an activity.

NOTE: If you assign this Flow Action to an Assignment that is one of the possible paths in the Process, for example, a path that results from the Decision shape, you need to consider the likelihood value of the Assignment. If this Flow Action has the highest likelihood value at run time, the form for this

©2025 Pegasystems Inc

> Flow Action appears as the default form when your application displays a Perform harness, and the system also runs the Data Transform and the activity. As a result, users interact with the updated values on a form. For more information, see Types of decision logic and Changing the decision logic in a Process.

> ⚠ **IMPORTANT:** You can configure pre-processing only for Flow Actions that do not support bulk-processing. Bulk processing cannot use a pre-processing activity, Data Transform, or a robotic Automation.

For relevant training materials, see the Flow Action processing module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. In the list of Flow Action instances, open the Flow Action that you want to edit.
4. On the **Action** tab, in the **Pre-processing** section, in the **Apply Data Transform** field, specify a Data Transform that the system applies to the current primary page at run time.
5. If the Data Transform includes parameters, in the **Parameters** section, enter values for the parameters.
6. In the **Run activity** field, specify an activity that the system runs before performing any additional processing for the Flow Action.

   Consider the following factors when you use an activity for pre-processing:

   - This activity runs only once, when a user selects the Flow Action for an Assignment, even if the application displays the Action area more than once. For example, when the data that the user submits for the first time fails validation. The activity also runs when you preview the Flow Action.
   - In the activity that you want to use, do not perform a Commit operation, and do not transfer the Assignment.

©2025 Pegasystems Inc

7. If the activity has parameters, in the **Parameters** section, enter values for the parameters.

   For example:

   To allow users to return control to a completed Assignment that is marked as an entry point, use the *jumpToTask* activity from the *Work-* class, and then complete either the **TaskName** parameter with the internal shape name, such as `ASSIGNMENT52`, or select the **previousEntryPoint** checkbox. For an example, see the standard Flow Action entitled *Work-.Previous*.

8. If you use Robotic Desktop Automation (RDA), specify the pre-processing Automation that you want to run on the user's desktop to retrieve data to display to users on a form in your application:
   a. In the **Run robotic Automation** field, enter an Automation name.
   b. In the **Description** field, enter text that describes what the Automation does.

   NOTE: Automations always run synchronously and pause the workflow. After an Automation completes processing, the system updates the Pega Clipboard, and renders the section that the Flow Action references.

9. Click **Save**.

What to do next:

To make the Flow Action available for reuse in App Studio, for example, while reusing the Flow as a smart shape, mark the Flow Action as a relevant record. For more information, see Marking a record as relevant.

Processes can be reused in App Studio as smart shapes, encapsulating advanced and complex functionalities in a low-code element that developers and business representatives can incorporate into their Case Types in App Studio. For more

PEGA ©2025 Pegasystems Inc

> information about Studios interoperability and modular reuse, see Reusable elements in App Studio.

Related tasks

- Configuring a Data Transform as a pre/post processing for a flow action

Configuring post-processing for a Flow Action

Define the behavior of your application after users complete an Assignment by providing a post-processing configuration for a Flow Action. With a post-processing configuration you can save time, automate your processes, and take advantage of additional options that improve manipulating data in a Case. Because you can use a Data Transform, an activity, and a robotic Automation, your application development process is flexible and can help you meet your individual requirements. For example, you can use savable Data Pages to save information that the current Case does not include. Consider a scenario in which a user who has an account in an online shop edits personal details when placing a shipping order. Although the modification of the contact information is external to the main Case, a user can still save the relevant data.

To store modified information, you can use a top-level Data Page that you need to provide in the Flow Action. You can also use an embedded Data Page. However, in this scenario the new information persists only in the context of a current Case and your application does not save the new data to any external system.

For post-processing, the system first invokes a robotic Automation, saves the Data Pages, runs the Data Transform, and then runs an activity. Specify the elements of post-processing according to your needs. For example, you can specify only robotic Automation and a Data Transform, and you can omit providing an activity.

For relevant training materials, see the Flow Action processing module on Pega Academy.

1. In the navigation pane of Dev Studio, click **Records**.

©2025 Pegasystems Inc

2. Expand the **Process** category, and then click **Flow Action**.
3. If you use the Robotic Desktop Automation (RDA), specify the post-processing Automation to run on the user's desktop when you want to send data that users enter in a Case form to all the applications that run on the user's desktop:
   a. In the **Run robotic Automation** field, enter an Automation name.
   b. In the **Description** field, enter text that describes the purpose of the Automation.

> **NOTE:** Automations always run synchronously and pause the workflow. After an Automation completes processing, the system updates the Pega Clipboard, and then the Data Transform runs.

4. **Optional:** To save a Data Page after the Flow Action completes, in the **Savable Data Pages** subsection, list the Data Pages that you want to save:

Choices	Actions
Use a Data Page	a. Click **Add Data Page**. b. **Optional:** To save the Data Page only under certain circumstances, in the **When** field, enter a When Condition Rule. At run time, if the When Rule evaluates to true, the application saves the Data Page. If you leave this field blank, the application always saves the Data Page. c. In the **Data page** field, specify a Data Page to save.

©2025 Pegasystems Inc

Choices	Actions
	For more information, see Saving data in a Data Page as part of a Flow Action.
Use a data relationship by providing an autopopulated property	a. Click **Add Data Page**. b. **Optional:** To save the Data Page only under certain circumstances, in the **When** field, enter a When Condition Rule. At run time, if the When Rule evaluates to true, the application saves the Data Page. If you leave this field empty, the application always saves the Data Page. c. Select the **Use associated property** checkbox. d. In the **Autopopulate property name** field, specify an autopopulated property.

5. **Optional:** To add more Data Pages, click **Add Data Page**, and then repeat step 4.
6. **Optional:** To monetize efforts connected with performing the Flow Action, in the **Apply cost** field, specify the Flow Action cost:
   - To specify the cost manually, enter an expression or a number in units of any type, such as currency.
   - To calculate the value automatically, enter a property reference that identifies a numeric property that is used in your computation.

   In a single application, use only numbers or property references.
7. In the **Apply Data Transform** field, specify a Data Transform that the system applies to the current primary page at run time.
   For more information, see Data Transforms.

©2025 Pegasystems Inc

8. In the **Run activity** field, enter the activity that the system runs after processing the Flow Action is complete.
   Consider the following factors when you use an activity for post-processing:
   - Do not include a Commit method in the activity.
   - If you use an activity that depends on parameter values that a user enters after selecting this Flow Action, enter the activity in the post-processing configuration rather than placing the activity in the Process.
   - You can use the activity to perform validation processing and add messages to the Pega Clipboard if validation fails. Validation by applying an activity is more suitable for complex validation scenarios. The activity can also perform additional processing that occurs after the user successfully submits the Flow Action.

   For more information, see Creating an activity and Setting SLAs for ad hoc Assignments.

9. If the activity includes parameters, in the **Parameters** section, enter values for the parameters.
10. Click **Save**.

What to do next:

To make the Flow Action available for reuse in App Studio, for example, while reusing the Flow as a smart shape, mark the Flow Action as a relevant record. For more information, see Marking a record as relevant.

Processes can be reused in App Studio as smart shapes, encapsulating advanced and complex functionalities in a low-code element that developers and business representatives can incorporate into their Case Types in App Studio. For more information about Studios interoperability and modular reuse, see Reusable elements in App Studio.

PEGA ©2025 Pegasystems Inc

Related tasks

- Preventing validation errors in cascading drop-down lists
- Configuring a Data Transform as a pre/post processing for a flow action

Configuring back-to-back processing for a Flow Action

Accelerate Case resolution by allowing users to perform more Assignments in a Case after processing of the current task is complete. When an application routes multiple Assignments to a single user, performing the Assignments in sequence increases efficiency and promotes uninterrupted workflow.

Back-to-back processing is optional but considerably increases the productivity and efficiency of your system. For example, a flow might require the editing and sending of three items of outgoing correspondence. In the workflow, a Split Join shape might cause the system to create three correspondence editing Assignments, all in a single Worklist. After a user edits the first correspondence item and completes one of the three Assignments, the user can immediately continue processing the remaining two Assignments.

Configuring back-to-back processing is a part of activity post-processing. Post-processing defines which actions users can take after they complete a task in a Case.

After you configure back-to-back processing, the system finds and displays only Assignments that the current user is qualified to perform and for which the value of the *pyActionTime* property from the *Assign-* class is past.

1. In the navigation pane of Dev Studio, click Records.
2. Expand the Process category, and then click Flow Action.
3. In the **Post-processing** section, in the **Back-to-back processing configuration** subsection, select the Look for an Assignment to perform checkbox.

©2025 Pegasystems Inc

> **Result:**
>
> At run time, the system searches for another Assignment for the same Case on the user's Worklist.

4. **Optional:** To ensure a continuous workflow, specify system behavior when the user's Worklist does not contain any relevant Assignments:
   - To search for open Assignments in the user's Worklist from other Processes that are active for the same Case, select **If not found, look for Assignments in other flows on this Case**.
   - To search for open Assignments from the parent Case of the current Case, select **If not found, look for Assignments in flows on the parent Case**.
   - To search for Assignments in Work Queues, select **For each also consider Assignments in Work Queues**

     The search scope expands to examine the Assignments in the Work Queues of the user's team and Assignments on the user's Worklist.

     Optionally, your application can use additional information for the detailed search criteria that the system uses when you select this checkbox. The standard decision tree *Assign-Workbasket.PerformCriteria* defines these criteria. For more information, see Decision trees.

5. **Optional:** To define what content is displayed to users after all of the Assignments are complete, select the appropriate behavior in the **If an Assignment is not being performed** list:
   - To display a refreshed harness form for the current Case, select **Show Harness**, and then in the text field, enter the relevant harness.

     For example, you can use a standard post-Assignment `Confirm` harness that displays the results of processing or computations based on user input.

©2025 Pegasystems Inc

> **NOTE:** Avoid specifying a harness of the `Perform` type that displays the Case in edit mode.

- To configure the system to search for an Assignment by using the Get Next Work logic, select the **Get Next Work**.

  The system uses information about the user's skillset and Urgency of available Assignments to calculate the next Assignment for the user to process. For more information about the Get Next Work algorithm, see Automated and intelligent work assignment with Get Next Work and Defining work routing settings for an operator.

- To configure the system to select the next Assignment in the workpool that is assigned to the current user, select **Get Next Work in Current WorkPool**.
- To close the user form and return to the previous work area, such as the Process Work home view, select **Close the Work Object**.

> **NOTE:** This setting applies only to **Submit** and **Next/Finish** Action buttons. For more information about Action buttons, see Configuring Action buttons for a Flow Action.

6. Click **Save**.

What to do next:

To make the Flow Action available for reuse in App Studio, for example, while reusing the Flow as a smart shape, mark the Flow Action as a relevant record. For more information, see Marking a record as relevant.

Processes can be reused in App Studio as smart shapes, encapsulating advanced and complex functionalities in a low-code element that developers and business

PEGA ©2025 Pegasystems Inc

> representatives can incorporate into their Case Types in App Studio. For more information about Studios interoperability and modular reuse, see Reusable elements in App Studio.

Configuring Action buttons for a Flow Action

Specify how users can interact with an Assignment in a case by defining buttons on a Flow Action. To meet your unique business needs, assign your own custom labels to the buttons so that users know exactly what Action takes place after clicking the button.

Consider a scenario in which a user places an order in an online shop. After the user provides their shipping details, a review form with the order details opens. Custom Action buttons help the user return to the editable form to change their details. The labels also communicate that in the next Step in the Process, the user chooses a payment method.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. In the list of Flow Action instances, open the Flow Action that you want to edit.
4. On the **Action** tab, in the **Action buttons** section, define buttons with which users can interact:

Choices	Actions
Hide default Action buttons	Select the **Hide the default Action section buttons** checkbox. If you select this checkbox, ensure that you provide custom navigation, such as buttons, drop-down menus, and links, in the body of the Action section. For example, use the *pxButton* control and a Cancel Action instead of the **Cancel** Action button. If you do not provide an

©2025 Pegasystems Inc

Choices	Actions
	alternative navigation method, users cannot proceed with a Case. For more information, see Configuring a Button control .
Customize the text on the default navigation buttons	a. Select the **Customize the Action section buttons labels** checkbox. b. In the field of a relevant Action button, enter your custom text. You can customize the following Action button labels: • **Submit button label** that submits the current work item. This button typically only appears in the last Step of a Flow Action. • **Previous button label** that displays the previous Step in the Flow Action, if one exists. This button only appears after the user moves past the first Step. • **Cancel button label** that cancels the Process and closes the Flow Action without saving any changes. • **Next button label** that displays the next Step in the Flow Action, if one exists. This

©2025 Pegasystems Inc

Choices	Actions
	button does not appear in the last Step of the Flow Action. • **Save button label** that saves the current form without submitting. This button is not displayed in the first and the last Steps of the Flow Action. **NOTE:** The Save Action button is supported only for applications based on Constellation. **For example:** The following figure shows customized button labels: *Flow Action buttons*
Configure when the Save **Action button is visible**	Choose when the **Save** Action button is visible by selecting a corresponding option from the drop down list:

©2025 Pegasystems Inc

Choices	Actions
	• Always - to make the Save Action button visible for every Step except for the first and the last. • Never - to hide the Save Action button for every Step. • When - to make the Save Action button visible according to the provided When Rule. **NOTE:** The Save Action button is supported only for applications based on Constellation.

5. Click Save.

Result:

At run time, users interact with buttons that have the customized, descriptive labels, as in the following figure:

©2025 Pegasystems Inc

Custom buttons on an order form

What to do next:

To make the Flow Action available for reuse in App Studio, for example, while reusing the Flow as a smart shape, mark the Flow Action as a relevant record. For more information, see Marking a record as relevant.

Processes can be reused in App Studio as smart shapes, encapsulating advanced and complex functionalities in a low-code element that developers and business representatives can incorporate into their Case Types in App Studio. For more

©2025 Pegasystems Inc

> information about Studios interoperability and modular reuse, see Reusable elements in App Studio.

Configuring indicator for a Flow Action

To design Cases that precisely meet your business requirements, define the results that user interactions with a Case have by placing a Flow Action at a relevant place in a Process. With Flow Actions, users can update information in an Assignment, resolve work, or both.

Consider a scenario in which a customer service representative (CSR) submits a form with a loan request from a customer. You can decide whether the CSR only updates information in a Case or whether the CSR can also resolve the Assignment and move the Case to the next task. An indicator on an Assignment or a connector between Assignments determines where in a Process that Flow Action happens.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Flow Action**.
3. In the list of Flow Action instances, open the Flow Action that you want to edit.
4. On the **Action** tab, in the **Indicator** section, in the **Used as** list, specify how you intend to use the Flow Action:
   - To update the Assignment and the current Case, select **Local Action**.

     By performing local Actions, users can update, but not resolve, an Assignment. The local Action retains the Process at the same Assignment shape, so that the current user, or another user in some Cases, can choose another Flow Action. Users can invoke local Flow Actions from a modal window or from a Flow Action form.

   - To update the current Case, complete the Assignment, and advance the Case, select **Connector Action**.

©2025 Pegasystems Inc

Typically, users invoke connector Flow Actions from a Flow Action form. Local Flow Actions are alternatives to autogenerated controls.

- To use the Flow Action as both a local Action and a connector Action, select **Local and Connector**.

5. **Optional:** To configure when the Flow Action is available for bulk processing, select one of the following options:

   Disqualify this Action from bulk processing
   Excludes the Flow Action from the list of Flow Actions that are available for bulk processing. The Action is available on the Full Page View in a Case. For more information, see Configuring a Full Page View.

   Run this Action only from bulk processing
   Includes the Flow Action only in the list of Flow Actions that are available for bulk processing. The Action is unavailable on the Full Page View in a Case. For more information, see Configuring a Full Page View.

   Apply pre-processing Data Transform
   Uses a Data Transform to automatically fill the form when the Flow Action is used in bulk processing.

   You can perform bulk processing of Flow Actions in the following situations:

   - When the Flow Action does not have circumstances.
   - For Assignments in various Cases, when the *Applies To* class of the Flow Action is a class that is an ancestor to all the Cases. For more information about classes, see Understanding class hierarchy and inheritance.

     By default, some standard Flow Actions do not support bulk processing because the Process can involve only one Case at a time. To change the default setting, copy the Flow Action to your Ruleset, and then make the change. For more information, see Copying a Rule or data instance.

6. Click **Save**.

©2025 Pegasystems Inc

What to do next:

To make the Flow Action available for reuse in App Studio, for example, while reusing the Flow as a smart shape, mark the Flow Action as a relevant record. For more information, see Marking a record as relevant.

Processes can be reused in App Studio as smart shapes, encapsulating advanced and complex functionalities in a low-code element that developers and business representatives can incorporate into their Case Types in App Studio. For more information about Studios interoperability and modular reuse, see Reusable elements in App Studio.

Adding optional Actions to paused Processes

Complement a paused Process with an optional Action to allow Case workers to accomplish related work while they wait for a Process to resume. For example, a customer service representative who processes a job application can review and update personal details of a job applicant before other child Cases, such as collecting required documents, reach a resolution.

Before you begin:

- Add a Wait shape to your Case Lifecycle. For more information, see Pausing and resuming Processes in Cases.
- Add a Local Flow Action . For more information, see Flow Actions.

1. In the **Case Lifecycle** section, open a Process that contains a Wait shape by clicking Configure Process.
2. Click Open Process, and then, in the Process diagram, double click the Wait shape.
3. In the **Details** section, select the Users can choose to continue Process checkbox.

©2025 Pegasystems Inc

4. In the **Local Action permitted for end user** section, in the Select local Flow Action field, press the Down arrow key, and then select an Action that you want to add.
5. **Optional:** To add more local Actions, click Add local Action, and then repeat step 4.
6. In the **Advanced wait options** section, in the Assigned to workbasket field, press the Down arrow key, and then select a Work Queue that your team can access.
   If you do not provide a Work Queue, your application routes the Assignment for the optional Action to an internal Work Queue, which is not associated with a team.
7. **Optional:** To select a section that displays the Action, in the Harness name field, press the Down arrow key, and then select a section that you want to use.
8. **Optional:** To change the message that is displayed when a user opens the Assignment for the optional Assignment, press the Down arrow key in the Instructions field, and then select a field value.
9. **Optional:** To assign a Service-Level Agreement to the Action, in the Service level field, press the Down arrow key in the Instructions field, and then select a Service-Level Agreement Rule.
10. Click Submit.
11. In the upper-right corner of the Case Type, save your changes by clicking Save.
12. **Optional:** To view your changes, switch to App Studio, and then click Preview and run the Case.

> ⚠ **IMPORTANT:** Avoid using Save and run to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

PEGA ©2025 Pegasystems Inc

Result:

When a Case reaches the Wait Step in the Lifecycle, users can perform the optional Assignment or manually resume the Process.

- Flow Actions
- Pausing and resuming Processes in Cases
- Assignment shapes in Processes

Managing Case Types

Reflect your individual business needs and requirements by modifying your Case Types. You can rename, import, or delete a Case Type. For example, if your business requirements change and one of your Case Types is no longer relevant, you can delete that Case Type.

- **Importing a Case Type**
- **Renaming Case Types**
- **Deleting Case Types and Cases**
- Case Types

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.

PEGA ©2025 Pegasystems Inc

Before you begin:

Add an application that contains Case Types that you want to reuse to your built-on applications stack. Ensure that the application is on top of the stack. For more information, see Adding built-on applications.

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** pane, click **Options** > **Import Case Types**.
4. Review the list of 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 checkbox 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 Process for a Case Type, continue to reside in the built-on application. You can leave them there, or create specialized versions in your application.

- Importing a Case Type
- Case Types

©2025 Pegasystems Inc

Extension points for Case Type creation

You can use extension points to customize the processing that occurs when you add or import Case Types to your application. For example, you can copy records from built-on applications to reuse resources.

The following extension points are supported by the New Application wizard and the Case Type Explorer:

- *@baseclass.pyIsFrameworkAssetGenerationForCasesEnabled*
- *@baseclass.pyPrepareFrameworkAssetsForGenerationSetup*
- *Rule-Obj-CaseType.pyPrepareFrameworkAssetsForGeneration*
- *@baseclass.pyPrepareFrameworkAssetsForGenerationPost*

> ⚠ **IMPORTANT:** Do not use the **Obj-Save** or **Commit** methods in your overridden versions because it can lead to unexpected results.

All records that are created by your extension points are saved to the database by an activity that is higher in the call stack.

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 **Homeowners Insurance Claim** Case Type to **Car and Homeowners Insurance Claim**.

> ⚠ **IMPORTANT:** When you rename a Case Type by updating its label through the Case Type field, this action changes only the display name of the Case Type. The underlying class name, which was established when the Case Type was

PEGA ©2025 Pegasystems Inc

initially created, remains unchanged. This distinction is important because the class name is integral to the system's architecture and dictates how Cases are processed and managed within the application.

For more information, see Creating a Case Type Rule form in Dev Studio.

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.

   For example:

   Enter `Car and Homeowners Insurance Claim`, as shown in the following figure.

Renaming a Case Type

3. In the upper-right corner of the Case Type, save your changes by clicking **Save**.

©2025 Pegasystems Inc

4. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Case Types

Deleting Case Types and Cases

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 Cases saves you time and money through the efficient management of your database space.

1. In the navigation pane of App Studio, click **Case Types**.
2. In the row of the Case Type that you want to delete, choose the Scope of your change:
   - To delete the Case Type and all associated assets, click **More options** > **Delete Case Type**.
     The assets that are associated to a Case Type include a Data Model, Case Lifecycle, and the UI elements.

> ⚠ **CAUTION:** Deleting a Case Type also deletes all Case instances associated with that Case Type. This action is irreversible without a backup, and you will permanently lose any data contained within the open Cases. Ensure that you thoroughly review and confirm that

©2025 Pegasystems Inc

> you no longer need any of the Case instances before you proceed with the deletion of a Case Type.

- To delete all Case instances, attachments, Pulse conversations, and Assignments for the Case Type, click **More options** > **Delete Cases**.

3. Click **OK**.

   The following figure shows how to delete a Case Type from a list of Case Types in App Studio:

Deleting a Case Type

- Case Types

Automated and intelligent work assignment with Get Next Work

Each time a Case worker completes an Assignment and does not have an immediate next Assignment, Get Next Work can pick the most urgent Assignment and prompt the

©2025 Pegasystems Inc

Case worker to complete the highest priority Assignment. By automating the selection of Assignments with the intelligent algorithm, you ensure that top-priority work is always processed first, avoid cherry-picking, and avoid multiple users working on the same Assignment.

The Get Next Work functionality selects the next Assignment for a user based on criteria such as urgency of the work, user authority, and skills. For example, if a Case worker has skills in credit card compliance, the Get Next Work function assigns that Case worker to handle the most urgent unassigned Assignment that requires that skill. You can easily customize the criteria by which Get Next Work selects work for users based on your business needs.

The Get Next Work functionality retrieves Assignments from all the applications to which the user has access.

Run-time experience

Case workers can also use the Get Next Work functionality on the Case level in a User Portal when you turn on the feature in the settings of the Case Type.

With the feature on, the **Get Next Work** link is displayed on a banner when a Case worker has no more immediate Assignments to complete in the current Case, as shown in the following figure:

Sample Case showing the Get Next Work banner and link

The Case worker can click the **Get Next Work** link, and, depending on the configuration of the operator record for the user, the Get Next Work algorithm searches the Worklist

©2025 Pegasystems Inc

or Work Queue of the user to provide a single Assignment that meets all other next work criteria, such as work urgency or user's skills. For more information, see Providing the Get Next Work link in a Case.

Additionally, Case workers can use the Get Next Work functionality when it is available on a landing page. When you enable the functionality, a Case worker can click the **Get Next Work** button on a landing page in a User Portal to open the top-priority Assignment from the Worklist or Work Queues of the Case worker. For example, in the following figure, the **Get Next Work** button is available in the upper-right corner of the Home page of the User Portal:

The Get Next Work button on the Home page

For more information, see Enabling Get Next Work button on a landing page.

Routing logic setup

The Get Next Work logic works on the Case level and on the landing pages of the application when there are no open Assignments linked to a Case worker.

By default, the next work item is picked from the Worklist of the user first, as shown in the following figure. Get Next Work picks the top Assignment on the Worklist of the user, and accesses a Work Queue only if the Worklist of the user is empty. Based on the configuration of the operator record, the feature first searches the Worklist or the Work

©2025 Pegasystems Inc

Queues that are listed on the operator record. If the Get Next Work logic finds an Assignment, the *GetNextWorkCriteria* decision tree further examines it.

Get Next Work logic

Pega Platform™ includes an option to trigger the Get Next Work algorithm to iterate through the Work Queues of the user first with the **Get from Work Queues first** option, which is available in the operator record linked to the user. For more information, see Defining work routing settings for an operator.

When multiple Work Queues are listed on a user's operator record, the Get Next Work logic processes the Work Queues from top to bottom. The following figure shows the area in an operator record in which the system lists the Work Queues in a certain order:

©2025 Pegasystems Inc

Work Queues list in a sample operator record

If you configure an **Urgency Threshold** for the Work Queue, then the Get Next Work logic prioritizes the Assignments with Urgency above the defined threshold, as shown in

©2025 Pegasystems Inc

the following figure. The logic considers lower Urgency Assignments only after all applicable Work Queues are empty of Assignments with Urgency above the threshold.

Urgency Thresholds in a sample operator record

Suppose we have three Work Queues with different Urgency Thresholds: `Loan Approvals` with a threshold of 75, `Repayment Extensions` with 90, and `Complaints Approvals` with 95. The Get Next Work algorithm adheres to the following order to fetch the highest-priority Assignment from the Work Queues:

1. The algorithm starts by looking for an Assignment with the highest Urgency in the `Loan Approvals` Work Queue with an Urgency of 75 or higher.
2. If no Assignment meets the 75+ threshold in the `Loan Approvals` Work Queue, the algorithm then checks the `Repayment Extensions` Work Queue and selects the highest-Urgency Assignment with an Urgency of 90 or higher.
3. If there are no Assignments meeting the 90+ threshold in the `Repayment Extensions` Work Queue, the algorithm proceeds to the `Complaints Approvals` Work Queue and retrieves the highest-Urgency Assignment with an Urgency of 95 or higher.
4. If there are no Assignments in the `Complaints Approvals` Work Queue that meet the 95+ threshold, the algorithm goes back to the `Loan Approvals` Work Queue and fetches the highest-Urgency Assignment available there.

This sequence ensures that the algorithm prioritizes Assignments based on their Urgency Thresholds, moving from one Work Queue to another as needed to find the most critical Task.

©2025 Pegasystems Inc

You can also set the Get Next Work algorithm to retrieve Assignments from all the Work Queues that correspond with the Team that you provided in the operator record. The **Use all Work Queue Assignments in user's team** checkbox in the operator record for the user enables this logic. This option also hides the **Work Queues** and the **Urgency Threshold** lists, because the system does note use these lists. For more information, see Defining work routing settings for an operator and Fields for operator teams, Work Queues, and schedules.

If you select **Merge Work Queues** checkbox in the operator record for the user, the algorithm consolidates Assignments from all of the Work Queues of the user, sorts the Assignments by Assignment Urgency, and then returns the most urgent Assignment in any Work Queue. This option also hides the **Urgency Threshold** list. If you already selected the **Use all Work Queue Assignments in user's team** checkbox, the system automatically selects the **Merge Work Queues** checkbox and you cannot bypass the consolidating option. For more information, see Defining work routing settings for an operator and Fields for operator teams, Work Queues, and schedules.

If you configure the Assignment to route by using the standard *ToSkilledWorkbasket* routing activity, then the logic considers the skills that are defined on the operator record of the user when retrieving the next Assignment. An Assignment can have both required and desired skills. The Get Next Work functionality considers required skills only. For more information, see Configuring custom routing logic for Assignments, Defining work routing settings for an operator, and Fields for operator teams, Work Queues, and schedules.

For more information about the Get Next Work algorithm and relevant training materials, see the Get Next Work feature article and the Designing Get Next Work challenge on Pega Academy.

- **Enabling Get Next Work button on a landing page**
- **Get Next Work customization**

©2025 Pegasystems Inc

Related concepts

- Operators
- Assignment shapes in Processes
- Intelligent routing

Related tasks

- Assigning tasks to users

Enabling Get Next Work button on a landing page

Show users the next most important Assignment by configuring the **Get Next Work** button to display on a landing page of your application. For example the My Work landing page, which provides quick access to the user's Cases and Assignments.

> **Before you begin:**
>
> Prepare the configurable components:
>
> - Create a landing page that uses the column template. For more information, see Creating a landing page for an application.
> - Create the Views and Insights that you want to add to the landing page. For more information, see Creating Views and Creating Insights.

The **Get Next Work** button can be added to the following major types of landing pages:

- A column-based landing page
- A list-based landing page
- A dashboard landing page

For more information, see Landing pages.

1. Open the page that you want to edit.

©2025 Pegasystems Inc

a. In the navigation pane of App Studio, click **Channels**.
b. In the **Current Channel interfaces** section, click the icon that represents the Portal with the page that you want to edit.
c. On the Portal page, on the **Content** tab, click the **Landing pages** category.

> **For example:**
>
> Click `My Work`.

d. In the **Landing pages** pane on the left side of the screen, click the page that you want to edit.

2. In the property pane on the right, select the **Enable Get Next Work** checkbox.

> **Result:**
>
> The following figure shows the **Get Next Work** button enabled for the My Work landing page of the User Portal:

The Get Next Work button on the My Work landing page

> When you click the **Get Next Work** button on the Home page, the next prioritized Assignment opens for processing.

Related concepts

©2025 Pegasystems Inc

- Automated and intelligent work assignment with Get Next Work

Related tasks

- Configuring a column-based landing page
- Configuring a list-based landing page
- Configuring a dashboard
- Providing the Get Next Work link in a Case

Get Next Work customization

You can adjust the Get Next Work algorithm to the specific needs of your application and your business operations.

For example, you can review and override the *Assign-.GetNextWorkCriteria* decision tree to add or modify criteria that are appropriate to your operations. The criteria may take into account daily or weekly cutoff events. Also, if your algorithm for finding open Assignments differs in major ways from those provided by standard Rules, you can override the following standard activities with others that are specific to each work pool or work type:

- *Work-.findAssignmentInWorkBasket*
- *Work-.findAssignmentInWorklist*
- *Work-.getContentForGetNextWork*

IMPORTANT: Customization of the Get Next Work feature is suitable for developers with an advanced understanding of Pega Platform™ development. Ensure that you understand the Get Next Work algorithm, Rules operations, and Application Settings.

- For more information about the Get Next Work feature, see Automated and intelligent work assignment with Get Next Work, Get Next Work feature, and the Designing Get Next Work challenge on Pega Academy.

©2025 Pegasystems Inc

- For more information about Application Settings, see Configuring production-level run-time settings and Defining values for an application setting.

Overview of entry points for customization

The core workings of the Get Next Work feature are encapsulated in the *Work-.getNextWorkObject* activity, which is a final Rule. The main entry point is the *Work-.GetNextWork* activity, which is meant to be called directly from the user interface and is involved with the display of the Assignment (or lack of Assignment). *Work-.GetNextWork* first calls the *getNextWorkObject* activity to find the right Assignment, and then calls other activities to display the Assignment, as specified by the steps in the following figure:

Processing instructions for the Work-.GetNextWork activity

The *Work-.GetNextWork* activity uses the following parameters:

Input Parameters

- *TargetFrame*: passed to the *Work-.Perform* primary output activity.
- *currentWorkPool*: retrieves Assignments only from the current work pool.

©2025 Pegasystems Inc

- *WorkBasket*: helps ignore the Operator ID settings and focus only on retrieving the most urgent Assignment from the provided Work Queue.

Output Parameters

- *AccessGroupName*: specifies to which Access Group you want to switch. An optional, rarely used parameter - the caller does not need to check the Access Group at all, because the *Work-.GetNextWork* activity handles switching Access Groups as needed.

The *Assign-.GetNextWorkCriteria* Decision tree is also available for customizations of the Get Next Work logic. You can use the decision tree to filter the results returned by the *GetNextWork* activity. For example, you can define and use custom When Rules in the decision tree or, if necessary, create circumstanced versions of the decision tree.

> ⚠ **IMPORTANT:** Using the *GetNextWorkCriteria* decision tree for filtering can impact performance because items are iterated and opened one by one. Always ensure that the *GetNextWork* Rule performs the main filtering.

Besides circumstancing the *GetNextWork* activity, it is also possible to circumstance the *GetNextWorkCriteria* decision tree for a particular Work Group. There are other alternatives, too, such as specializing in the *getContentForGetNextWork* activity to call a different Decision Rule to get the expected results.

Application Settings related to Get Next Work

You can define your own Get Next Work functionality by modifying the following Application Settings Rules:

GetNextWork_SkipSkillCheck

If you know that your application does not require skills or uses them rarely, set this to `enabled` so that the SQL skips all the complicated JOINS on the operator and Assignment skill tables.

©2025 Pegasystems Inc

GetNextWork_SkipSkillCount

By default, if an Assignment lists more than one skill, the user must possess all of these skills in order to be able to receive the Assignment. When you set this to `enabled`, it changes the logic to any of these skills. Changing this setting to `enabled` also improves performance because the query has to perform fewer comparisons.

GetNextWork_SkipUnskilledAssignments

By default, an operator that has skills is entitled to receive either the Assignments for which they have the skill(s) to perform, or Assignments that do not specify any skills at all. However, you may want your skilled workers to focus on receiving Assignments that require their skills. If you want this functionality, set the Rule to `enabled`. This is a setting to override in a production Ruleset and set to `disabled` when the unskilled operators are being overwhelmed, so that the skilled operators can pitch in, Kanban-style.

GetNextWork_MoveAssignmentToWorklist

This is the default way of enabling iteration through a Work Queue. Every time you click **Get Next Work**, it pulls an Assignment out of the Work Queue and into your Worklist. However, sometimes you may want to keep the Work Queue Assignments right where they are. Disabling this Rule does so and introduces an additional JOIN on the system locks table to ensure that the Assignment that you are performing is not locked by any other user (which otherwise would be a major performance problem due to lock contention.)

GetNextWork_WorkBasketUrgencyThreshold

This Rule defines the default Urgency Threshold if you leave the Urgency Threshold field blank in the Operator ID configuration.

The default value is set to `0` so this setting has no impact. However, in a scenario in which the value in the Rule is set to `51` and the Operator ID specifies Work Queues `A`, `B`, and `C` (where `B` is set to threshold `76` but the others are left blank), then the system first looks for Assignments with Urgency `51` to `100` in Work Queue `A`, then Urgency `76` to `100` in Work Queue `B`, and then Urgency `51` to `100` in Work Queue `C`. If none are found, the system looks at Urgency `0` to `50` in Work

©2025 Pegasystems Inc

Queue A, Urgency 0 to 75 in Work Queue B, and finally Urgency 0 to 50 in Work Queue C.

Configuration options for Get Next Work

You can specify the main Get Next Work settings on the Work tab of the Operator ID form and modify the logic of the *GetNextWorkCriteria* decision tree.

The Get Next Work settings on the Operator ID form include the following options:

Get from Work Queues first

If this setting is not selected, the system first gets the highest Urgency Assignment in your Worklist.

Merge Work Queues

When this setting is selected, the system ignores the Urgency Threshold column because all of the Assignments in the listed Work Queues are equally viable. The system retrieves all of the Assignments in all of the listed Work Queues, sorted by Urgency.

Work Queue

Work Queues that the system checks for the next Assignment, in the order listed.

Urgency Threshold

As explained in the *GetNextWork_WorkBasketUrgencyThreshold* Application Setting, this option introduces a second pass concept.

For example, with this Application Setting applied, assuming the Urgency Threshold in the Application Setting is set to 0, the Get Next Work algorithm runs a query for AccountException Assignments with Urgencies 0 to 100. The system gets the first 500 records. If none of the AccountException Assignments are selected, the algorithm runs a query for the first 500 in the *AdminProtocolWB* Work Queue between Urgencies 51 to 100. If none of the 51+ AdminProtocolWB Assignments yield results, the algorithm runs a query for the 0 to 50 in the *AdminProtocolWB* Work Queue Assignments.

©2025 Pegasystems Inc

Use all Work Queue Assignments in user's team

Available only when you first select Merge Work Queues. This setting hides the entire Work Queues list because it compares the Assignment's team to the current user's team.

You finish specifying Get Next Work settings by determining whether the Assignment the system has retrieved is viable or not. For that, after opening the Assignment, you pass it to the *GetNextWorkCriteria* decision tree:

Default configuration of the GetNextWorkCriteria decision tree

The *GetNextWorkCriteria* decision tree can use the following *When* Rules:

IsReadyToBeWorkedOn

If the Assignment has an SLA defined, the top section of the SLA Rule determines whether an Assignment is ready or not, as shown in the following figure:

©2025 Pegasystems Inc

The Assignment readiness settings in a sample SLA Rule

HasBeenUpdatedByMeToday

Useful when a user does not want to return to an Assignment that they have already done some work on in a given day. This is a typical iteration through a Worklist: the user clicks **Get Next Work**, gets an Assignment from their Worklist, realizes they do not want to complete the task today, performs a local action on the Assignment, and then if they click **Get Next Work** again, the system does not retrieve that Assignment because the user has already updated it today.

> **NOTE:** The user needs to select **Save** as a local action. Clicking **Save** in the application does not work.

CurUserHasRequiredSkills

This check is entirely redundant with what the initial query does, unless the *GetNextWork_SkipSkillCheck* Application Setting is set to `enabled`.

©2025 Pegasystems Inc

Guidelines for successful customization

- Ensure that the *pxUrgencyAssign* Assignment Urgency is set to a meaningful value because it is a basis for sorting and prioritization.

  Make the Urgency values meaningful in all Case Types that send Assignments to the same Work Queue unless the **Merge Work Queues** or **Use all Work Queue Assignments in user's team** checkboxes are selected on the operator record. 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 Case Types that send Assignments to the same Work Queue (unless the **Merge Work Queues** option or the **Use all Work Queue Assignments in user's team** checkboxes are selected on the operator record).

  For example, in some business settings, it may be appropriate to complete every Assignment in the `CustomerOrder` application before any Assignments in the `Researching` application. Also, it does not make sense for the `Police Report` Case Type to use Urgency in the 40-60 range, and the `Fire Report` Case Type to use Urgency in the 80-100 range if both Case Types send Assignments to the Dispatch Work Queue, because a Case worker would only ever get a Police Report if there are absolutely no Fire Report Assignments.

  By default, a standard Declare Expression Rule computes the Assignment Urgency value based on work item Urgency and other criteria. For example, you can override the Urgency value using criteria such as dollar amounts, customer categories, specific Assignments that are often on the critical path, work item age, work types, and more, as the basis for determining Assignment Urgency.

- If an Assignment has skills that are marked 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.

PEGA ©2025 Pegasystems Inc

> **NOTE:** In Processes, you can use the standard router activity *ToSkilledWorkBasket* to send an Assignment with required or wanted skills to a Work Queue.

> **TIP:** Instead of adding a large number of individual skills, embrace the concept of skill packages that combine related skills into broader categories, enhancing efficiency while minimizing skill-related slowdowns. For instance, creating a skill package such as `Expert System Operator` can encompass multiple related skills, simplifying skill management and optimizing query performance.

- If Work Queues are searched, review the Work Queues for each user to ensure that the order in which they are listed on the Work tab of the Operator ID form is correct.
- To identify a minimum cutoff value for Assignment Urgency, override the *GetNextWork_WorkBasketUrgencyThreshold* Application Settings Rule.

  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 are 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.

- During testing and when in production, note the situations in which the algorithm that you have implemented does not return any Assignments.
- Do not list a Work Queue in a user's settings for which they do not have the skills to complete most of the Assignments.

Related concepts

©2025 Pegasystems Inc

- Decision trees
- Intelligent routing

Related tasks

- Configuring custom routing logic for Assignments
- Creating an activity
- Defining work routing settings for an operator

Related reference

- Activity form - how to create activities for Processes
- Fields for operator teams, Work Queues, and schedules

Related information

- Alternative ways to find work

Enabling bulk processing of Cases in Constellation

Provide users with the ability to create and process multiple Cases simultaneously by configuring bulk operations. For example, they can adjust Service-Level Agreement times and add a note to all selected Cases at the same time. This feature is valuable in applications that handle large volumes of Cases, which can assist businesses in saving time, ensuring consistency, and reducing errors across various processes.

Bulk processing helps users handle repetitive tasks efficiently by enhancing productivity across different sectors. For example, the feature can apply in the following use cases:

- Instant batch approval of expense reports in financial services, handling up to 30 reports simultaneously.
- Customer support teams can manage support tickets by efficiently assigning 100 new tickets to team members simultaneously.

PEGA ©2025 Pegasystems Inc

- Human resources departments can streamline onboarding by assigning tasks to 10 new hires simultaneously.
- Inventory management teams can update product information, such as adjusting prices for 250 products in a single Action.

> **NOTE:** Bulk processing is limited to 250 items.

Constellation applications support bulk operations on Case records in tables and lists that are displayed on landing pages and Case pages.

Set up overview:

Setting up bulk processing in a Constellation application consists of the following major steps:

1. Adding an Action to the Case Lifecycle.

   This configuration occurs in App Studio in a Case Type form.

2. Creating a landing page to display a list of Cases and the bulk operations option.

   This configuration occurs in App Studio on a landing page form.

3. Enabling bulk processing for the selected Action.

   This configuration occurs in Dev Studio in a local Flow Action Rule that powers the Action that you add to the Case Lifecycle.

Set up in detailed steps:

Adding an Action to the Case Lifecycle

1. Add an Action to your Case Type:
   - To capture new or updated data from users, add the Collect information type of Action.

©2025 Pegasystems Inc

- To enable manual intervention, such as transitioning a Case or applying a status change without necessarily collecting additional data. add the User Action type.

For more information, see Adding optional Actions to Cases.

> **NOTE:** If you add a Stage-only Action, all the Cases for which you want to perform a given bulk operation must be in the Stage to which the Action was added.

For example:

In the **Case-wide Actions** list, add a Collect information Action, and then change the default Action name from `Case-wide local Actions ([number])` to `Add info`, as shown in the following figure:

New Case-wide Action added to the Case Type

2. On the Case Type form, click **Save**.

Creating a landing page

3. Create a one-column landing page in your Portal.
   For more information, see Creating a landing page for an application.

©2025 Pegasystems Inc

For example:

Create a landing page called `Bulk processing page`.

4. Add the default Case list View to the landing page.

NOTE: Every Case Type includes a View that lists all instances of the Cases that belong to a given type. By default, the name of that View is [Case Type name] List.

For more information about configuring the landing page, see Configuring a column-based landing page

For example:

In this scenario, the View with the list of Cases is `Evidence gathering List`. The following figure shows the Case list View added to the landing page:

©2025 Pegasystems Inc

Landing page with a default Case list View

5. Open the Case list View by clicking the link in its name, and then in the **Edit View** pane, select the **Allow bulk Actions** checkbox.
   For more information about configuring Views, see Configuring list Views.
6. Add the landing page to the menu of your Portal.
   For more information, see Defining a menu for a Constellation Portal.
7. On the Portal form, click **Save**.

Enabling bulk processing for the Flow Action Rule

8. In the header of your development studio, click the **Switch Studio** menu, and then click Dev Studio.
9. In the navigation pane of Dev Studio, click **App**.
10. On the **Classes** tab, in the Application Explorer, expand the node named after your Case Type, and then expand the **Process** node to access your Flow Action:

©2025 Pegasystems Inc

- To open a Flow Action of the Collect information type, click **Flow Action** > **CasewideLocalActions[number]**.
- To open a Flow Action of the User Action type, click **Case Type** > **pyDefault**, and then select the Action form the list.

For example:

The Action that you add in this scenario, CasewideLocalActions1, is in the **Evidence gathering** > **Process** > **Flow Action** node, as shown in the following figure:

Location of the CasewideLocalActions1 Flow Action Rule

11. On the Flow Action Rule form, on the **Actions** tab, in the **Bulk processing** section, clear the **Disqualify this Action from bulk processing** checkbox.

©2025 Pegasystems Inc

12. **Optional:** To make the Flow Action available only for bulk processing, select the **Run this Action only from bulk processing** checkbox.
13. **Optional:** To apply pre-processing and post-processing to the Flow Action, to the right of the **Apply pre-processing data transform field**, click the **Open** icon, and then define the parameters for the operations.
    For more information, see Configuring pre-processing for a Flow Action and Configuring post-processing for a Flow Action.
14. On the Flow Action form, click **Save**.

Result:

The landing page includes a table that supports the **Add info** bulk Action:

Bulk Action configuration

Related concepts

©2025 Pegasystems Inc

- Landing pages
- Flow Actions

Related tasks

- Creating Cases from tables

Changing Case Statuses

Communicate to stakeholders how a business process moves towards resolution, by changing the Case Status in the Lifecycle. For example, you can assure the stakeholders that the Case advances within the time frame.

You can configure the status change when the Case reaches a certain Stage or Step.

For relevant training materials, see the Case Status module on Pega Academy.

> **Before you begin:**
>
> Configure Stages and Steps:
>
> 1. Create Stages for your Case Lifecycle. For more information, see Stages in a Case Lifecycle.
> 2. Create Steps for the Stages. For more information, see Steps in a Case Lifecycle.

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 **Lifecycle**.
3. In the **Case Lifecycle** section, define when the Case changes the status:

©2025 Pegasystems Inc

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. In the upper-right corner of the Case Type, save your changes by clicking Save.
5. **Optional:** To view how your changes are reflected at run time, click Preview and run the Case.

> ⚠ **IMPORTANT:** Avoid using Save and run to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, the Case changes the status when the Case reaches a point that you define in the Case Lifecycle. The following figure shows an example of setting a Case Status in a Stage:

PEGA ©2025 Pegasystems Inc

Setting a Case Status in a Stage

- **Case Status values**
- **Defining a Case Status**
- **Standard Rules and extension points for Case Status**

Related concepts

- Case Status values

Related tasks

- Defining Conditions for skipping a Stage
- Configuring a Case resolution
- Defining a Case Status

Related reference

- Best practices for defining Case Statuses

©2025 Pegasystems Inc

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, the `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.

PEGA ©2025 Pegasystems Inc

Case Status	Description
`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.

NOTE: 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 Case Status.

NOTE: As a best practice, use descriptive, business-oriented values for Case statuses, so that different stakeholders across your enterprise can understand the business meaning of each status. Typically, naming Case statuses is within the responsibility of business representatives.

- Standard Rules and extension points for Case Status
- Assigning tasks to users
- Assigning users automatically at run time with business logic
- Completing work on time
- Creating and managing Cases
- Creating Cases from the navigation pane of a Portal
- Reopening resolved Cases
- Best practices for defining Case Statuses

PEGA ©2025 Pegasystems Inc

Defining a Case Status

Define a custom Case Status to more precisely communicate the progress that a Case makes toward resolution.

1. In the header of Dev Studio, click **Configure** > **Case Management** > **Processes** > **Status Values**.
2. Review the list of Case Statuses, and ensure that a Case Status that meets your business needs does not already exist.
3. Create a field value that defines your custom status:
   a. In the header of Dev Studio, click Create > **Data Model** > **Field Value**.
   b. In the **Label** field, enter the text for the custom status value.

      As a best practice, use one of the following prefixes to support standard reports and maintain continuity in between your custom status and standard status values: `New-`, `Open-`, `Pending-`, or `Resolved-`. Use only alphanumeric symbols in Case Statuses.

      > **NOTE:** As a best practice, use descriptive, business-oriented values for Case statuses, so that different stakeholders across your enterprise can understand the business meaning of each status. Typically, naming Case statuses is within the responsibility of business representatives.

   c. In the **Field Name** field, enter **pyStatusWork**.
      At run time, your application maps a value of the *pyStatusWork* property with a Case Status.
   d. In the **Context** section, select an application layer, applies to class, and Ruleset based on how you plan to reuse the status value in your application.

©2025 Pegasystems Inc

> **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

Best practices for defining Case Statuses

Follow best practices to ensure that Case Statuses are clear and convey the current state of your business processes. Use consistent Case Statuses and formats to efficiently communicate the progress and status of both resolved and unresolved Cases in your application.

Preliminary considerations

Before you define values for your Case Statuses, consider the following questions:

- What progress do you want to report on while work is ongoing?
- What variants of Case Type resolution do you need to communicate?
- How do you want to group Cases by status?
- What are the major events that might trigger a status change?

Use the following best practices to define Case Statuses according to the above conditions:

Apply intentional alphabetical order of prefixes

Case Statuses have a prefix and a suffix in their names by default, for example `Open-Ready`, where `Open` is the prefix and `Ready` is the suffix.

©2025 Pegasystems Inc

The default prefixes include `New`, `Open`, `Pending`, and `Resolved`. These alphabetically and logically ordered prefixes are intentional and help simplify the sorting of Cases by status. As a best practice, use the recommended prefixes for your Case Status values across all Case Types.

Use prefixes and suffixes effectively

Several features in Pega Platform™ rely on the prefix of a Case Status. You can use the Case Status to filter Cases based on their prefix.

In the following example, you create a report that lists Cases resolved by a user that excludes Cases with a `Resolved-Cancelled` status because they represent work that is not resolved by a user, as shown in the following figure:

Case Status as a Condition in a report

All closed Cases should have a status with the `Resolved` prefix. The following sample configuration of a report uses a Condition that states that a Case Status does not start with the letter R. Applying this Condition avoids showing any resolved Cases in the report.

©2025 Pegasystems Inc

A report that filters out all the resolved Cases

A hyphen and a meaningful suffix often follow the Case Status prefixes and indicate the disposition. For example, `Resolved-Approved` indicates that a Case is in a resolved status and has been approved.

> **NOTE:** If you need to create new prefixes or remove them altogether, consider the change very carefully. Creating new prefixes might require a refactoring procedure. As a best practice, extend existing prefixes with new suffixes instead of creating new prefix values. You might also want to explore field values as a way to translate the status name. By using field values, you avoid the need to refactor statuses for features that use them. For more information, see Creating Field Values.

Limit the number of Case Statuses

You can create as many statuses as necessary to achieve your business goals, but, as a best practice, define no more than ten statuses for a given Case Type. Keeping the

©2025 Pegasystems Inc

number of statuses lower than ten helps users better understand how a Case progresses by avoiding multiple statuses.

Use consistent Case Statuses

Define consistent Case Statuses across Case Types to keep business processes logical across your application. In this way, users can move from one Case Type to another and carry over their knowledge of Case Statuses without the need for additional training.

Related tasks

- Creating Field Values
- Changing a Stage with open Assignments

Related reference

- Standard Rules and extension points for Case Status

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.

PEGA ©2025 Pegasystems Inc

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, Process shape, or Case Lifecycle 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 automatic conditional creation of child Cases
- Defining a Case Status

Changing a Stage with open Assignments

Create a flexible application that meets your business needs even in untypical scenarios by configuring a Case to change Stage without resolving open Assignments in a current Stage. As a result, users can complete a specific Assignment at any time during a Case Lifecycle.

Consider the example of a recruitment Process. The Review Stage might include running a background check that an HR worker might need to additionally perform later on in a Case, after the reviews of the documents from the candidate and the score from the job interview.

By default, when a Case moves to the next or another Stage, an application resolves open Assignments in the current Stage. For example, in a recruitment Case, when your

PEGA ©2025 Pegasystems Inc

Case moves from the Review to Offer Stage, the application resolves open Assignments in the Review Stage. If you want to keep Assignments open, configure Stage transition by using the Utility shape, instead of typical shapes, to move a Case further in a Case Lifecycle.

As a best practice, create a Process that stores the Utility shape, and then add the Process in an appropriate Stage in your Case Lifecycle.

1. Create a Process with the Utility shape:
   a. In the header of Dev Studio, click **Create menu** > **Process** > **Flow**.

      > **NOTE:** "Flow" is a legacy name for a Process.

   b. In the **Label** field, briefly describe the purpose of the Process.

      > **For example:**
      >
      > Enter `Change Stage with open work`.

   c. In the **Context** section, select an application layer to store the Process.
   d. **Optional:** To change the default identifier for the Process, click **Edit**, and then provide a unique value in the **Identifier** field.
   e. In the **Apply to** field, click the Down arrow key, and then select a class that stores the Process.
   f. In the **Add to Ruleset** list, select a Ruleset and a Ruleset version to store the Process.
   g. Click **Create and open**.
2. On the **Diagram** tab, remove the default Assignment shape by selecting the shape and clicking **Delete**.
3. On the toolbar, click **Flow Shapes**, and then click **Utility**.
4. Connect the **Utility** shape by dragging the connector end points to the connection points on the Start and End shapes.

©2025 Pegasystems Inc

Result:

The following figure shows a Process with a Utility shape connected to the Start and End shapes:

Utility shape in a Process

5. Open the dialog box by double-clicking the **Utility** shape.
6. **Optional:** To provide a unique name for the shape, in the **Utility properties** dialog box, in the **Utility** field, enter a new name.

For example:

Enter `Change Stage`.

7. In the **Automation details** section, in the **Select type of Rule** list, select **Activity**.
8. In the **Rule** field, enter `pxChangeStage`.
9. In the **Parameters** section, define the Stage that the Case enters from the Utility shape:
   - To move the Case to the next Stage, select the **ChangeToNextStage** checkbox.
   - To move the Case to another Stage in the Case Lifecycle, in the **ChangeToStage** field, enter a Stage ID or Stage name.

TIP: You can configure the Utility shape to move the Case to the next Stage by selecting the **ChangeToNextStage** checkbox, and then providing another Stage that the Case enters when **ChangeToNextStage** returns a false or invalid value.

©2025 Pegasystems Inc

10. **Optional:** To provide additional information for auditing reasons, in the **AuditNote** field, enter the audit note to add to the Case history when the Case moves to the Utility shape.
11. Leave the Assignments open after the Case moves to another Stage by clearing the **CleanUpProcesses** checkbox.
12. Define properties to capture the run-time output details:
    a. In the **CurrentStageLabel** field, enter a label for the current Stage to display at run time.
    b. In the **CurrentStage** field, enter the ID for the current Stage.
    c. In the **AutomationErrors** field, enter a page of the *Code-Automation-Errors* class that stores the details if the *pxChangeStage* activity fails during processing.
13. Click **Submit**.

What to do next:

Integrate the Process that stores the Utility shape with your Case Lifecycle, and decide whether to launch it every time, or only conditionally:

- To start the Process with the Utility shape every time, see Adding a sequential Process to a Stage.
- To start the Process with the Utility shape only conditionally, see Adding a sequential Process to a Stage and Conditionally starting a Process.

- Processes in a Case Lifecycle
- Adding a parallel Process to a Stage
- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools
- Building Case Types

PEGA ©2025 Pegasystems Inc

Completing work on time

Enforce the Service-Level Agreements (SLAs) that you define 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.

The purpose and components of SLAs

A Service-Level Agreement (SLA) is a that defines a goal and deadline, which are intervals of time that you can apply to a 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 a work item, you define the suggested time required to resolve work. When you define a deadline, you define the ultimate time in which to resolve work. You can also define escalation Actions that your application performs when the goal and deadline expire.

In a sample scenario, the goal for an Assignment is three days and the deadline is five days. An application starts calculating the goal and deadline after the Assignment is ready for processing. If a Case worker fails to complete the Assignment in three days, an escalation Action occurs and the worker receives a notification about an elapsed goal. If the Case worker fails to complete work within the next two days, another escalation Action takes place and the manager of the Case worker receives a notification about a passed deadline. Finally, the worker completes the Assignment. The following figure shows the process:

©2025 Pegasystems Inc

SLAs in Case processing

To help Case workers prioritize work, each Assignment includes an Urgency field, which is a numeric value that indicates the importance of the Assignment. You can decide how Urgency increases after a goal or deadline elapses to bring attention of workers to the tasks that are top priority and need immediate processing.

For more information, see Urgency and Adjusting initial Urgency for an Assignment.

Types of SLAs

You can apply SLAs to the following elements of Case processing:

Case

SLA processing starts when a user starts or reopens a Case, and stops when the Case reaches resolution.

For more information, see Setting Service-Level Agreements for Case resolution.

©2025 Pegasystems Inc

Stage

SLA processing starts when a Case enters the Stage, and ends when the Case leaves the Stage. You can apply SLAs to a Primary Stage and an Alternate Stage, and to optional Stages.

For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

Process in a Stage

SLA processing starts when a Process starts, and ends when the last Step in the Process finishes. You can apply SLAs to Processes that are part of a Case Lifecycle and to optional Processes.

For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

Assignment

SLA processing starts when a user or an Automation creates or routes an Assignment to a Work Queue or Worklist so that the Assignment is ready for processing, and ends when the Assignment is completed or stops due to an error condition. The time at which a user opens the form for an Assignment has no effect on the SLA.

For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

Approval Step

SLA processing starts after a user receives the approval Step that is ready for processing, and ends when the Step is completed.

For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

PEGA ©2025 Pegasystems Inc

Background processing of SLAs

Pega Platform™ supports the following ways of background processing for SLAs:

An agent

By default, the system uses the agents infrastructure to process the SLA items. The class in which the system stores items for agent processing of SLAs is *System-Queue-ServiceLevel*.

A queue processor

For better scalability, ease of use, and faster background processing, you can enable a queue processor for background processing of SLAs in your application. The class that the system uses to store items for the queue processor processing of SLAs is *System-Message-QueueProcessor-DelayedItem*.

For more information, see Queue processing for SLAs.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **Urgency**
- **Setting Service-Level Agreements for Case resolution**
- **Setting Service-Level Agreements for Stages, Processes, and Steps**
- **Setting SLAs for ad hoc Assignments**
- **Adjusting initial Urgency for an Assignment**
- **Creating a Service-Level Agreement Rule in Dev Studio**

PEGA ©2025 Pegasystems Inc

- **Configuring delayed service-level processing**
- **Defining an escalation Action for a Case**
- **Defining an escalation Action for a Stage and Process**
- **Defining an escalation Action for an incomplete Assignment**
- Automating work by creating Case Types
- Completing work on time - Pega Academy module

Urgency

Urgency is represented by a numeric value, ranging from 0 to 100, that worklists and Cases display to indicate importance. Use Urgency to bring visibility to unresolved Assignments and Cases in your application that need immediate action from Case workers. As a result, you ensure that Cases reach resolution within the timelines that you define with your stakeholders.

Utilize Urgency to standardize the way Case workers resolve Cases or Assignments in an application. The higher the Urgency value, the more important it is to address that item. Urgency can be calculated or updated in the following ways:

- Using a Service-Level Agreement (SLA) Rule. For more information, see Creating a Service-Level Agreement Rule in Dev Studio.
- Using standard Flow Actions that are provided by Pega Platform. For more information, see Flow Actions.

The default Urgency value is 10. You can define how elapsed goals and deadlines affect the Urgency of Cases, Stages, Processes, Assignments, and approval Steps directly in the Case Lifecycle Designer.

For example, a loan request Case that includes an Assignment to review the banking history of a customer who applies for a loan has a goal of three days. If the customer service representative (CSR) fails to meet the goal, the Assignment Urgency increases by

PEGA ©2025 Pegasystems Inc

20. With an initial Urgency of 10, the total Urgency is then 30. The following figure shows an example of the configuration of this Urgency increase:

Increasing Urgency

If the CSR misses the deadline for the Assignment, which is two days in this sample scenario, the Urgency rises again, by 30, and totals 60. The following figure presents how the Urgency rises:

©2025 Pegasystems Inc

Urgency changes throughout Assignment processing

There are two types of Urgency:

- Case Urgency
  - A value between 0 and 100 that defines the priority of the Case.
  - The *Work-.pxUrgencyWork* standard property holds the Case Urgency value and the value is computed by using a standard Declare Expression.

©2025 Pegasystems Inc

    - The initial Urgency of a Case is calculated by taking various values into consideration such as the values set in the SLA Rule form, default Urgency set during Case creation, Stage SLA, and Process SLA values.
- Assignment Urgency
    - A value between 0 and 100 that defines the priority of completing an Assignment.
    - The *Assign-.pxUrgencyAssign* standard property holds the Assignment Urgency value and the value is computed by using a standard Declare Expression.
    - The initial Urgency of an Assignment is calculated based on the Case Urgency value.

Additionally, in Dev Studio, you can adjust initial Urgency when you create a Service-Level Agreement Rule. You can then apply the Rule to a Case, a Stage, a Process, or a Step. You can use this option to prioritize certain work in your application, for example, Tasks that concern VIP customers or that are critical to your organization. You can also use the following Rules to change Urgency:

Work-.UrgencyUpdate

A Flow Action that you can call during Case processing to re-calculate the Urgency. An application adds the integer that you provide to the current Urgency of the Case.

> **NOTE:** Ensure that you provide different values for Case and Assignment Urgency.

Work-.pyUrgencyWorkAdjust

A property that you can set to increase or decrease the current Urgency of a Case. The *Work-.pxUrgencyWork* Declare Expression evaluates the value that you provide.

©2025 Pegasystems Inc

> **NOTE:** If you update the Case Urgency, the system uses the new value to calculate the Assignment Urgency for new Assignments, but not for the existing Assignments.

Assign-.pyUrgencyAssignAdjust

A property that you can set to increase or decrease the current Urgency of an Assignment. The *Assign-.pxUrgencyAssign* Declare Expression evaluates the value that you provide.

At run time, the system updates Urgency each time the value of a standard property changes, or a user saves a Case.

- Completing work on time
- Setting Service-Level Agreements for Case resolution
- Setting Service-Level Agreements for Stages, Processes, and Steps
- Creating a Service-Level Agreement Rule in Dev Studio

Setting Service-Level Agreements for Case resolution

Encourage customer service representatives (CSRs) to resolve Cases on time and enforce your Service-Level Agreements (SLAs) by setting goals and deadlines for a Case Type. As a result, you save time and resources because you avoid missing deadlines. SLA measurement starts when the Case starts and ends when the Case finishes.

For example, if a CSR of an insurance claim request does not resolve the request within a week, the Urgency of this Case rises and the manager of the CSR receives a notification.

For relevant training materials, see the Completing work on time module on Pega Academy.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **Goal & deadline**.

©2025 Pegasystems Inc

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 SLA, select **Existing SLA**. In the auto-complete field, enter the SLA 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**.

PEGA ©2025 Pegasystems Inc

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.

For example:

The following image shows an SLA being set for a Case with a goal and deadline. The goal is set for two days with an Urgency increase of ten, and a default notification message sent to the creator. The deadline is set for five days with an Urgency increase of thirty, and also has a default notification message sent to the creator. See the following image for an example of this configuration:

Setting an SLA for a Case

©2025 Pegasystems Inc

> **NOTE:** If you update the Case Urgency, the system uses the new value to calculate the Assignment Urgency for new Assignments, but not for the existing Assignments.

- Building Case Types
- Case Types
- Setting Service-Level Agreements for Stages, Processes, and Steps
- Defining an escalation Action for a Case
- Completing work on time

Setting Service-Level Agreements 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 customer service representatives to review loan requests within a goal of five days and a deadline of ten business days.

For relevant training materials, see the Completing work on time module on Pega Academy.

The system calculates goals and deadlines as the time between task assignment and task completion, unless your SLA states otherwise. For more information about advanced SLA configurations, see Creating a Service-Level Agreement Rule in Dev Studio.

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 **Lifecycle**.

©2025 Pegasystems Inc

3. In the **Case Lifecycle** section, click the element of a Case Lifecycle for which you want to define an SLA:
   - To define a goal and deadline for a Stage, click the Stage.

     > **NOTE:** You cannot set a goal and deadline for the Create Stage because the Create Stage collects data before Case processing starts. For more information, see The Create Stage.

   - To define a goal and deadline for a Process, click the Process.
   - To define a goal and deadline for a Step, click the Step.
4. In the properties pane, click **Goal & deadline**.
5. In the **Use Service-Level Agreement (SLA)** list, select the SLA criteria:
   - To define custom criteria for the goal and deadline, select **Custom SLA**, and then go to step 6.
   - To reuse criteria from an existing SLA, select **Existing SLA**. In the **Select** 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.

©2025 Pegasystems Inc

For example:

To indicate that the deadline 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 deadline elapses.

For example:

The following figure shows sample configuration of a goal and deadline for a Step. The goal is set to one day, after which Case Urgency increases by 10. The deadline is set to two days, after which Case Urgency increases by 30:

Setting SLAs for Stages, Processes and Steps

8. Click **Save**.

©2025 Pegasystems Inc

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 Process.

- Setting Service-Level Agreements for Case resolution
- Completing work on time

Setting SLAs for ad hoc Assignments

Set Service-Level Agreements (SLAs) on ad hoc Assignments to ensure that these additional tasks are completed within the necessary timeframes.

In Pega Platform™, ad hoc Assignments enable Case workers to introduce additional tasks into a Case at any time, enhancing flexibility in Case Management. For more information, see Creating ad hoc Assignments.

Before you begin:

Ensure that the **Create Assignments** optional Action is added to your Case Type. For more information, see Adding optional Actions to Cases and Adding optional Actions to Stages.

You set SLAs for ad hoc Assignments by modifying the *pySimpleCase* extension SLA Rule, which is referenced in a Data Transform in the activity that governs the behavior of ad hoc Assignments from the *pyCreateAdHocTasks* Flow Action in your application Ruleset.

PEGA ©2025 Pegasystems Inc

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Service Level Agreement**.
3. In the list of SLA Rule instances, click the *pySimpleCase* SLA Rule.
4. Click **Save as**, and then configure and save the SLA Rule in your application Ruleset by performing the actions in step 1 in Creating a Service-Level Agreement Rule in Dev Studio.
5. Configure the SLA settings that meet your business requirements by performing the actions in steps 2 through 6 in Creating a Service-Level Agreement Rule in Dev Studio.

   The following figure shows the configuration form of the extension Rule for setting SLAs for ad hoc Assignments:

©2025 Pegasystems Inc

The pySimpleCase extension SLA Rule

6. Click **Save**.

Result:

At run time, the ad hoc Assignments calculate and display deadline information, as shown in the following figure:

©2025 Pegasystems Inc

Run-time view of an ad hoc Assignment with deadline information

Related concepts

- Completing work on time
- Flow Actions

Related tasks

- Creating an activity

Adjusting initial Urgency for an Assignment

Communicate the priority of a given Assignment by adjusting initial Urgency. By configuring an Assignment Urgency, you provide a common understanding of task importance. As a result, you optimize workload management so that users can make informed decisions about which Assignment to process next. For example, in a Case Type for reviewing loan requests from VIP customers, you can increase the initial Urgency to ensure that a customer service representative (CSR) resolves such work as a priority.

©2025 Pegasystems Inc

Before you begin:

Assign a Service-Level Agreement to an Assignment. For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

Users can check an Assignment Urgency during run-time Assignment processing. The Urgency can increase during a Case Lifecycle when an Assignment passes a goal and deadline. The default Assignment Urgency is 10.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, click a Step that has a defined goal and deadline.
3. In the **Step** properties pane, on the **Goal & deadline** tab, open the SLA Rule by click the **Open** icon next to the Rule name.
4. In the **Start of service level** section, in the **Initial Urgency** field, define the initial Urgency by entering an integer between 0 and 99.
   An Assignment has a default Urgency of 10. The system adds the value in this field to determine the initial Urgency. For example, if you enter 10, the initial Urgency is 20.
5. Click **Save**.

What to do next:

Ensure that users notice and resolve Assignments with passed goals and deadlines. For more information, see Defining an escalation Action for an incomplete Assignment.

Related tasks

- Creating a Service-Level Agreement Rule in Dev Studio
- Assigning tasks to users

©2025 Pegasystems Inc

Related information

- Completing work on time

Creating a Service-Level Agreement Rule in Dev Studio

Enforce Service-Level Agreements (SLAs) and maintain timely resolution of Cases by defining the goal, deadline, and passed deadline time intervals by using an SLA Rule. For example, you can notify a manager about a passed goal and deadline, and then transfer the Assignment to another user. You can then reuse the setting for multiple Cases and Assignments, and as a result, manage time and resources efficiently.

As a best practice, and to take advantage of low-code tools, define SLAs in App Studio. Creating an SLA Rule in Dev Studio is recommended for advanced developers only. For more information, see Setting Service-Level Agreements for Case resolution and Setting Service-Level Agreements for Stages, Processes, and Steps.

1. Configure an SLA Rule:
   a. In the header of Dev Studio, click **Create menu** > **Process** > **Service Level Agreement**.
   b. In the **Label** field, enter a short description for the Rule.
   c. In the **Context** section, select an application to which the Rule applies.
   d. In the **Apply to (class)** field, select a class to which the Rule applies.
   e. In the **Add to Ruleset** field, select a Ruleset for the Rule.
   f. **Optional:** To enable traceability of the Rule, in the **Current work item** section, enter the work item to associate with the Rule, for example, the ID of a related bug.
   g. Click **Create and open**.
2. On the **General** tab, in the **Initial Urgency** field, define the Urgency by entering an integer between 0 and 99.
   An Assignment has the default Urgency of 10. The system adds the value in this field to arrive at the initial Urgency.

©2025 Pegasystems Inc

> **For example:**
>
> If you enter 10, the initial Urgency is 20.

3. In the **Assignment Ready** list, define the starting time for the Assignment:
   - To send the Assignment to a Worklist or a Work Queue immediately after the user or an automation creates an Assignment, select **Immediately**.
   - To define the starting time by using a Date/Time value, select **Dynamically defined on a property**, and then, in the **Get Date Time From** field, enter the property that references the value.
   - To start the Assignment after a period of time, select **Timed delay**, and then, in the **Days**, **Hours**, and **Minutes** fields, specify the delay.
4. In the **Service level definitions** section, in the **Calculate service levels** field, select a method of calculating intervals for goal and deadline:
   - To manually specify the intervals, select **Interval from when Assignment is ready**, and then, in the **Goal** and **Deadline** sections, enter the intervals.
   - To define the time intervals as values of properties, select **Set the value of a property**, and then, in the **Goal** and **Deadline** sections, in the **Property** fields, enter the properties that reference goal and deadline dates and times.

   The values that you can set are intervals of time that can be applied to a Case, Stage, Process, and Assignment:

   Goal
   Defines the interval for completing a task, which is typically measured from the SLA start time. The goal interval occurs once.

   Deadline
   The ultimate time within which the task should be completed before the Case or Step is considered to be late. The deadline is also measured from the SLA start time. The deadline interval occurs once.

5. **Optional:** To facilitate timely resolution of work, define escalation Actions that occur when the goal and deadline pass:

©2025 Pegasystems Inc

a. In the **Amount to increase Urgency** field, enter the number that an application adds to the initial Urgency then the time interval passes.

> **For example:**
>
> If the initial Urgency is 10, and then you configure the Rule to add 20 when the goal passes, the total Urgency is 30 after the goal passes.

b. In the **Actions** section, in the **Select Action** row, click **Edit**.

c. In the **Perform Action** list, select an Action that occurs when the time interval passes, and then provide relevant details if necessary.

> **NOTE:** Ensure that the Action that you select does not perform the *Commit* method because the commit occurs when SLA Rule processing ends. Providing an Action that performs the *Commit* might cause issues.

d. **Optional:** To perform the escalation Action only under specific circumstances, in the **When** field, enter a When Rule.

> **Result:**
>
> An escalation Action occurs if the When Rule returns a true value.

e. **Optional:** To add more escalation Actions, click **Add an Action**, and then repeat steps 5.a through 5.d.

> **TIP:** You can define escalation Actions for goal, deadline, and passed deadline events.

©2025 Pegasystems Inc

6. **Optional:** To define the number of the passed deadline events and the time when the passed deadline events occur, in the **Passed deadline** section, enter the appropriate values.

   For example:

   `Passed deadline` is a recurrent interval after the deadline. If you set **Passed deadline events** as 3, **Days** as 1, and **Increase urgency** as 10, the Urgency will increase by 10 every day after the deadline has passed, over a period of 3 days. After 3 days and 3 increases, there will be no further Passed deadline events.

7. Click **Save**.

What to do next:

To make the SLA Rule available for reuse in App Studio, mark it as a relevant record. AS a result, you can use the advanced Rule created in Dev Studio when you configure goals and deadlines in Case Types in App Studio. For more information, see Marking a record as relevant.

- Completing work on time
- Understanding and complying with Case Management best practices

Configuring delayed service-level processing

Provide solutions in scenarios when users need to wait before processing an Assignment by delaying Service-Level Agreements (SLAs). As a result, you deliver a flexible application and adjust processing to the unique business requirements of your customers.

For example, consider a scenario in which an employee of a trading company has a deadline of two days to review and approve a return of faulty goods from a customer. The deadline count starts when the employee receives the faulty goods, which happens

©2025 Pegasystems Inc

daily at 3:00 PM, even if the review Assignment in a complaint Case is ready earlier. In a different scenario, an employee needs to wait for two hours before processing a purchase order, because a customer can cancel finalization of the order for two hours after the purchase.

For relevant training materials, see the Delayed service-level agreement processing topic on Pega Academy.

> **Before you begin:**
>
> Assign an SLA to an Assignment. For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.

1. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
2. In the **Case Lifecycle** section, click a Step that has a defined goal and deadline.
3. In the **Step** properties pane, on the **Goal & deadline** tab, open the SLA Rule by clicking an icon next to the Rule name.
4. In the **Start of service level** section, define when an application begins the goal and deadline count by selecting an option from the **Assignment Ready** list:
   - To define the starting time by using a specific *DateTime* property, select **Dynamically defined on a property**, and then in the **Get Date Time From** field, enter the property that references the value.

     The *DateTime* property holds a precise starting time, for example, 3:00 PM on Monday August 27.

   - To start the Assignment after a specific period of time, select **Timed delay**, and then in the **Days**, **Hours**, and **Minutes** fields, specify the delay.
5. Click **Save**.

Related concepts

©2025 Pegasystems Inc

- Understanding and complying with Case Management best practices
- Steps in a Case Lifecycle
- Job schedulers and queue processors included with Pega Platform

Related information

- Completing work on time

Defining an escalation Action for a Case

Ensure that your business process reaches a resolution on time, by defining an escalation Action for a Case. By enforcing Service-Level Agreements (SLA) and implementing escalation Actions, you enforce timely resolution of Cases and meet deadlines that you agree on with your customers.

For example, if a customer service representative (CSR) fails to resolve a Case within first goal and then deadline, a reporting manager receives email messages with Case details.

For relevant training materials, see the Completing work on time module on Pega Academy.

> **Before you begin:**
>
> Add a custom SLA to a Case. For more information, see Setting Service-Level Agreements for Case resolution.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the Case working area, click the **Settings** tab, and then click **Goal & deadline**.
3. In the **Goal** section, in the **Notify** subsection, select recipients of the notification:
   - To notify a creator of the Case, select **Creator**.
   - To notify the participants of the Case, select **Participants**.

©2025 Pegasystems Inc

4. If you select to notify participants, click **Manage participants**, select the Case participants that you want to notify, and then click **Done**.
5. In the **Message** subsection, select a message type:
   - To send a standard notification, select **Use default message**.
   - To compose an email message, select **Compose a message**, provide a subject, click **Compose**, enter your message, and then click **Done**.
6. In the **Deadline** section, create a notification by repeating steps 3 through 5.
7. Click **Save**.

For example:

The following figure shows a Case with a Custom SLA including a two day goal. The time is calculated from the start of the Case, and the creator is notified with a custom message once the Case reaches the goal.

Defining escalation actions for a Case in the Settings tab

- Defining an escalation Action for a Stage and Process
- Setting Service-Level Agreements for Case resolution

©2025 Pegasystems Inc

Defining an escalation Action for a Stage and Process

Ensure that your business use Case moves forward in a timely manner by defining an escalation Action for a Stage and a Process in your Case Lifecycle. By notifying interested parties about an elapsed goal or deadline, you can keep the time frames that you agree on for your business use cases.

For example, if an HR worker fails to meet the goal for running a background check on a job candidate, a reporting manager receives a notification. If the worker misses the deadline, the manager receives an email message with relevant details.

For relevant training materials, see the Completing work on time module on Pega Academy.

> **Before you begin:**
>
> Define notification Channels. For more information, see Notification delivery Channels.

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 Lifecycle** section, click the element of a Case Lifecycle for which you want to define an escalation Action:
   - To send a notification after a goal or deadline for a Stage that elapses, click the Stage.

     > **NOTE:** You cannot set an escalation Action for the Create Stage because the Create Stage collects data before Case processing starts. For more information, see The Create Stage.

   - To send a notification after a goal or deadline for a Process that elapses, click the Process.

©2025 Pegasystems Inc

3. In the properties pane, click the **Goal & deadline** tab.
4. From the **Use Service-Level Agreement (SLA)** list, select **Custom SLA**.
5. 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**.
6. If you choose to notify participants, click **Manage participants**, select the Case participants that you want to notify, and then click **Done**.
7. 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**.
8. In the **Deadline** section, create a notification by repeating steps 5 through 7.
9. Click **Save**.

> **For example:**
>
> The following video shows a user configuring escalation Actions for a Loan Approval Stage. When the Stage reaches its goal, the creator is notified with a default message. When the Stage reaches its deadline, the customer and owner are notified with a default message.

> **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 Assignments routed to a user's Worklist and missed SLAs.

- Setting Service-Level Agreements for Stages, Processes, and Steps
- Defining an escalation Action for a Case
- Completing work on time

PEGA ©2025 Pegasystems Inc

Defining an escalation Action for an incomplete Assignment

Reduce the risk of unfinished work in a Case by defining an escalation Action, such as a notification to the manager. The manager receives the notification when an Assignment exceeds its goal or deadline.

For example, in a Case of reviewing a complaint for purchased goods, an application can notify a manager that a Case worker misses a goal for confirming customer details required to pay the reimbursement.

For relevant training materials, see the Completing work on time module on Pega Academy.

1. In the navigation pane of 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 **Lifecycle**.
3. In the **Case Lifecycle** section, click an Assignment with a custom goal and deadline.
   For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.
4. In the **Step** properties pane, click **Goal & deadline**.
5. 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**.
6. In the **Deadline** section, define an escalation Action by repeating step 5.
7. Click **Save**.

©2025 Pegasystems Inc

- Assignment shapes in Processes
- Collecting information from a user

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.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

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. As a best practice, create a Notification in App Studio using an automated Step for a simplified experience. For more information, see Sending event notifications from Cases.

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. For more information, see Notification delivery Channels.

©2025 Pegasystems Inc

At run time, the Channels can send notifications to a user in the following scenarios:

- Case processing reaches the **Send notification** Step.

  For more information, see Sending event notifications from Cases.

- A Pulse message is added to a Case or the **Home** page.

  For more information, see Configuring Pulse email notifications.

- An application routes an Assignment to the Worklist of a user, or a user transfers an Assignment to another user.

  For more information, see Sending email notifications about Assignments routed to a user's Worklist, Notifying Case participants about Case events, and Sending custom notifications about new Assignments.

- **Creating a Notification Rule**
- **Notification delivery Channels**

Creating a Notification Rule

Inform users about events in which they are involved or interested by creating a Notification Rule, which you can use, for example, to notify the Case workers about new Assignments or Case Status changes.

Notifications enhance operational efficiency by enabling timely communication, facilitating task management, and ensuring that users stay informed about critical events and actions.

Pega Platform™ includes standard Notification Rules that you can use in your application. You can use Records Explorer in Dev Studio to view all the Notification Rules available to you. The Notification Rules are part of the Process category. For more information, see Exploring Rules in your application.

PEGA ©2025 Pegasystems Inc

The system activates a Notification Rule when running the **Send notification** automated Step in a Case Lifecycle.

For more information, see Sending event notifications from Cases.

A Notification Rule determines how a notification is processed and delivered. The Rule consists of the following mandatory elements:

- A message that needs to be delivered
- Intended recipients
- Notification delivery channels

1. In the header of Dev Studio, click **Create menu** > **Process** > **Notification**.
2. In the **Notification Record Configuration** section, enter values in the fields to define the details of the Notification.
   For more information, see Creating Rules.

Defining a notification message

3. On the **Notification definition** tab, in the **Message** field, keep the default field value or enter a field value that contains your notification message.
4. If the field value that you provide in the **Message** field supports parameters, in the **Parameters** section, enter the parameters.
   For more information, see Defining the input parameters of a Rule.

Adding recipients of a notification

5. In the **Recipients** section, in the **Select context** field, specify the Data Page that is the source of property values for the recipients.
   For more information, see Data Page types, Data page reference, and Configuring page, page group, and page list properties.
6. In the **Recipient type** field, specify whether the recipient is internal or external:
   - To specify that the recipients are application users with valid operator IDs, select **Operator ID**.

©2025 Pegasystems Inc

- To specify that the recipients are external users, select **Email ID**.

7. In the **Recipient property** field, select the property on the Data Page that holds the value of the recipient *Operator ID* or *Email ID* property.
8. **Optional:** To add more recipients from the Data Page, click **Add property**, and then repeat steps 6 and 7.
9. **Optional:** To add recipients from another Data Page, click **Add recipients** and complete the **Recipients** section.

Configuring the delivery channels

10. Define the channels for delivering your notifications.
    For more information, see Notification delivery Channels.

Configuring Notification categories and specifying additional information

11. On the **Advanced** tab, in the **Category** field, keep the default value or enter a value of the *pyNotificationCategory* type.
    By default, a notifications widget is configured to display notifications of the `Default` category. If you use multiple widgets, the category that you specify in this field determines which widget displays the Default category notifications.

    For more information, see Configuring a Web gadget notification delivery Channel for Portal notifications and Setting notification preferences.

12. **Optional:** To prevent the Notification from being run when referenced, select the **Mute Notification** checkbox.

> **NOTE:** As a best practice, mute overridden Notifications rather than withdraw the Rules.
>
> For more information, see Setting Rule status and availability and Preparing for modifying Rules.

©2025 Pegasystems Inc

13. **Optional:** To prevent users from setting their own notification preferences at run time, select the **Not configurable from UI** checkbox.
14. If the configured channels require additional parameters at run time, in the **Additional information** section, pass the values of the parameters dynamically by storing them as key-value pairs.
    For more information, see Configuring additional parameters for notifications delivery Channels.
15. **Optional:** To provide information about what Clipboard pages the Notification Rule accesses or updates, on the **Pages & Classes** tab, specify the name and class of the pages.

> **NOTE:** Leave this tab blank if the other tabs of the Rule do not reference any pages. Complete this tab to describe the Clipboard context that you expect at run time. For more information, see Defining the pages and classes of a Rule and Differences between data pages and clipboard pages.

16. Click **Save**.

- Rule creation (Pega Academy topic)

Notification delivery Channels

A notification delivery Channel is a carrier that the application uses to notify users about events at run time. For example, users can receive email notifications when posts are made in Cases that they are following.

By using notifications effectively, organizations can ensure that users stay informed and engaged with critical updates and tasks within the Pega Platform™ environment.

You select and configure a notification delivery Channel for your notification on the **Channels** tab of a Notification Rule form.
You can configure your Notifications to use one or all of the following Channels:

©2025 Pegasystems Inc

Web gadget

Portal notifications are displayed as messages in the application at run time. You can include additional notification widgets to display different types of notifications separately.

For more information, see Configuring a Web gadget notification delivery Channel for Portal notifications.

Email

Notifications can be sent through email to users' registered email addresses. Users can reply to email notifications to post messages in a conversation.

For more information, see Configuring an email notification delivery Channel.

Mobile

For mobile-enabled applications, notifications can be delivered as push notifications to users' mobile devices. With mobile push notifications, users can approve and reject Cases, and respond to messages in Cases.

For more information, see Configuring a mobile notification delivery Channel and Engaging mobile users with push notifications.

If the existing Channels do not accommodate your requirements, you can also create custom notification channels. For example, banks send SMS messages to users to inform them about transactions in their accounts.

For information about creating a custom notification Channel, see Creating a notification Channel.

- **Configuring a Web gadget notification delivery Channel for Portal notifications**
- **Configuring an email notification delivery Channel**
- **Configuring a mobile notification delivery Channel**

©2025 Pegasystems Inc

- **Configuring additional parameters for notifications delivery Channels**
- **Displaying application-specific correspondence**
- **Creating a notification Channel**
- Creating a Notification Rule
- Setting notification preferences

Configuring a Web gadget notification delivery Channel for Portal notifications

Deliver notification messages directly to Portals to boost engagement and ensure that key information is readily accessible to users.

Pega Platform™ Portals provide a default notification widget for delivering notifications. You can configure this notification delivery Channel on the **Channels** tab of a Notification Rule form.

At run time, notifications are displayed as messages in a widget that you access by clicking the **Notifications** icon in the navigation pane of the Portal, as shown in the following figures:

©2025 Pegasystems Inc

Sample notification messages in a Portal

Sample Portal with the Notifications icon

For more information see Working with Portals and Organizing the main navigation for a standard work Portal.

> **Before you begin:**
>
> Create a Notification Rule. For more information, see Creating a Notification Rule.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Notification**.
3. Open the Notification for which you want to configure a Portal notification.
4. On the **Channels** tab, in the **Web gadget** section, in the **Display section** field, keep the default Section name or enter the name of a Section in which to display your notification message.

> **NOTE:** To ensure that users receive notifications from at least one Channel, you cannot disable Portal notifications in the **Enable web gadget** field. Users in traditional UI applications can update their notification preferences at run time. For more information, see Setting notification preferences.

©2025 Pegasystems Inc

5. In the **Default preference** list, set a default notification preference for all the users in your application.
   For more information, see Setting notification preferences.
6. **Optional:** To configure multiple notification widgets for different notification categories, perform the following actions:
   a. On the **Advanced** tab, in the **Category** field, enter a value of the *pyNotificationCategory* type.
      By default, a notifications widget is configured to display notifications of the `Default` category. If you use multiple widgets, the category that you specify in this field determines which widget displays default category notifications.
   b. **Optional:** To prevent the Notification from being run when referenced, select the **Mute Notification** checkbox.

      > **NOTE:** As a best practice, mute overridden Notifications rather than withdraw the Rules.

   c. **Optional:** To prevent users from setting their own notification preferences at run time, select the **Not configurable from UI** checkbox.
   d. If the configured Channel requires additional parameters at run time, in the **Additional information** section, pass the values of the parameters dynamically by storing them as key-value pairs.
      For more information see Configuring additional parameters for notifications delivery Channels.
7. Click **Save**.

- Configuring additional parameters for notifications delivery Channels
- Notification delivery Channels
- Engaging and notifying stakeholders
- Creating a Notification Rule

©2025 Pegasystems Inc

Configuring an email notification delivery Channel

Ensure prompt user action and improve collaboration across teams by configuring the Email Channel, which sends email notifications to alert users, for example about Case updates or Pulse messages posted on the Home page.

Before you begin:

- Create a Notification. For more information, see Creating a Notification Rule.
- Configure an email account. For more information, see Configuring outbound email.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Notification**.
3. Open the Notification for which you want to configure an email notification.
4. On the **Channels** tab, in the **Email** section, perform one of the following actions:
   - To customize the Email Channel, go to step 5.
   - To disable email notifications for all users, clear the **Enable email** checkbox, and then go to step 9.
5. Customize the Email Channel:
   a. In the **Subject** field, keep the default field value, or enter the field value that holds your notification message for email notifications.
   b. If the field value accepts parameters, in the **Parameters** section, enter the parameters for the field value.
   c. In the **Correspondence** field, enter a correspondence template for the email message body.
   d. In the **Default preference** list, enable or disable the email channel for all the users in your application:
   - To send email notifications instantly, select **Instant**.
   - To send email notifications daily, select **Daily**.
   - To send email notifications weekly, select **Weekly**.

©2025 Pegasystems Inc

- To stop sending email notifications, select **Disabled**.

> **NOTE:** Run-time users can update the preferences as needed. For more information, see Setting notification preferences.

6. **Optional:** To prevent the Notification from running when referenced, on the **Advanced** tab, select the **Mute Notification** checkbox.

> **NOTE:** As a best practice, mute overridden Notifications rather than withdraw the Rules.

7. **Optional:** To prevent users from setting their own notification preferences at run time, on the **Advanced** tab, select the **Not configurable from UI** checkbox.
8. If the configured Channel requires additional parameters at run time, in the **Additional information** section, pass the values of the parameters dynamically by storing them as key-value pairs.
   For more information see Configuring additional parameters for notifications delivery Channels.
9. Click **Save**.

Related concepts

- Notification delivery Channels

Related tasks

- Sending event notifications from Cases

©2025 Pegasystems Inc

Configuring a mobile notification delivery 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** checkbox 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**.

- Engaging mobile users with push notifications
- Configuring additional parameters for notifications delivery Channels
- Notification delivery Channels
- Engaging and notifying stakeholders
- Creating a Notification Rule

Configuring additional parameters for notifications delivery Channels

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

©2025 Pegasystems Inc

to notify users that follow a Case about a Pulse message in the Case, you 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.To pass the parameter to the Rule, in the **Key** field, provide a key to store the parameter name, for example, *PulseKey*, and then in the **Value** field, define the value for the specified key, for example, *PostMessage.pzInsKey*.

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 **Add a row** 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 delivery Channels
- Engaging and notifying stakeholders
- Creating a Notification Rule

Displaying application-specific correspondence

Help users more intuitively send correspondence from Cases by providing correspondence prompts that are specific to the context of the current application. As a result, run-time users interact with a relevant list of available correspondence templates, such as templated email messages, so that they can send correspondence faster.

©2025 Pegasystems Inc

You can mark correspondence as a top-level Correspondence Rule so that the application prompts this correspondence at run time. By default, applications prompt all top-level Correspondence Rules that are available for a Case through the class hierarchy.

To organize prompts in a more efficient way, create application-specific prompts. For example, in a banking application, you can create a list of correspondence templates that are specific to this application only.

1. On a Correspondence Rule form, mark the Correspondence as top-level:
   a. In the navigation pane of Dev Studio, click **Records**.
   b. Expand the **Process** category, and then click **Correspondence**.
   c. In the list of Correspondence instances, open the Correspondence that you want to edit.
   d. On the **Prompts** tab, in the **Top level** section, select the **This is a top-level Correspondence Rule** checkbox.

   > **Result:**
   >
   > Your application displays this correspondence at run time as a prompt.

2. Copy the *ListOfValidCorrs* activity into your application Ruleset:
   a. In the navigation pane of Dev Studio, click **Records**.
   b. Expand the **Technical** category, and then click **Activity**.
   c. In the list of activity instances, double-click the **ListOfValidCorrs** activity.
   d. In the form header, click **Save as**.
   e. In the **Context** section, in the **Apply to** field, enter your application class.
   f. In the **Add to Ruleset** list, select your application Ruleset and Ruleset version.
   g. Click **Create and open**.
3. Edit the *ListOfValidCorrs* activity:
   a. In the row of the *call LookupList* method, click **Expand to see method parameters**.

©2025 Pegasystems Inc

b. In the **Value** field of the `Action` parameter, enter `classgroup`, as shown in the following figure:

Editing ListOfValidCorrs activity

c. Click **Save**.

Result:

At run time, when users select a correspondence template, a list of prompts includes only top-level Correspondence Rules from the current application, as shown in the following example:

©2025 Pegasystems Inc

Correspondence prompts

- Notifying Case participants about Case events
- Sending event notifications from Cases
- Sending automatic emails from Cases

Creating a notification Channel

Expand your application by creating new notification channels to meet specific requirements, such as sending notifications as phone text messages. For example, a banking application can send SMS messages to clients when processing of their loan application reaches an important milestone.

For the scenario in which you want to create a Channel for sending SMS notifications, ensure that you have an SMS account defined in your application. For more information, see Creating an SMS account.

Extending the *Pega-Notification* class:

1. In the navigation pane of Dev Studio, click **Records**.
2. Select **SysAdmin** > **Class**.
3. Filter the **Class Name** column by `Pega-Notification`, and then open the *Pega-Notification* class.
4. Click **Save As**, and then, in the **Label** field, enter the name of your new Channel.

   For example:

   Enter `SMS Notification Channel`.

©2025 Pegasystems Inc

5. In the **Class Name** field, enter the name of the new class.

> **For example:**
>
> Enter `Pega-Notification-SMS`.

6. Click **Create and open**.
7. In the **Created in version** field, enter the version number, and then click **Save**.

Extending the *Data-Notification-Preference* class:

8. In the navigation pane of Dev Studio, click **Records**.
9. Select **SysAdmin** > **Class**.
10. Filter the **Class Name** column by `Pega-Notification`, and open the *Data-Notification-Preference* class.
11. Click **Save As**, and then in the **Label** field, enter the name of your new Channel.

> **For example:**
>
> Enter `SMS Notification Preference Channel`.

12. In the **Class Name** field, enter the name of the new class.

> **For example:**
>
> Enter `Data-Notification-Preference-SMS`.

13. Click **Create and open**.
14. In the **Created in version** field, enter the version number, and then click **Save**.

Updating the *pyGetNotificationChannelClass* map value Rule to map the new Channel name to the Notification Rule:

15. In the navigation pane of Dev Studio, click **Records**.
16. Select **Decision** > **Map value**.

©2025 Pegasystems Inc

17. Filter the **Map Name** column by `pyGetNotificationChannelClass`, and open the *pyGetNotificationChannelClass* map value.
18. Click **Save as**, and then click **Create and open**.
19. On the **Matrix** tab, click **Configure rows** > **Add row**, and then in the **Configure Rows** dialog box, in the field for the Channel name, enter the name for the new Channel.

    > **For example:**
    >
    > Enter `SMS`.

20. Click **Submit**.
21. On the **Matrix** tab, in the row with the new Channel name, in the **Preference Class** column, add the preference class that you created in steps 8 to 14.

    > **For example:**
    >
    > Enter `Data-Notification-Preference-SMS`.

    For more information, see Completing the Matrix tab and Map values.

22. In the **Run-time Class** column, add the class that you created in steps 1 to 7 as the new processing class.

    > **For example:**
    >
    > Enter `Pega-Notification-SMS`.

    For more information, see Completing the Matrix tab and Map values.

23. Click **Save**, and then check in the Rule.

    For more information, see Checking in a Rule.

Creating an activity for sending notifications in the new Channel:

©2025 Pegasystems Inc

24. In the header of Dev Studio, click **Create** > **Technical** > **Activity**.
25. In the **Label** field, enter the name of the record.

> **For example:**
>
> Enter `Send SMS Notification`.

26. In the **Apply to (class)** field, enter the class that you created in steps 1 to 7.
27. Click **Create and open** and then define your activity.
    For more information, see Creating an activity.

> **For example:**
>
> The following figure shows a sample configuration of the activity:

Sample configuration of an activity for sending SMS notifications

28. Click **Save**.

Extending the *Embed-NotificationChannel* class:

29. In the header of Dev Studio, click **Create** > **SysAdmin** > **Class**.
30. In the **Label** field, enter the name of the new Channel.

©2025 Pegasystems Inc

> **For example:**
>
> Enter `SMS Notification Channel`.

31. In the **Class Name** field, enter the name of the new class.

> **For example:**
>
> Enter `Embed-NotificationChannel-SMS`.

32. Click **Create and open** > **Save**.

Creating a Data Transform:

33. In the header of Dev Studio, click **Create** > **Data Model** > **Data Transform**.
34. In the **Label** field, enter `pyDefault`.
35. In the **Apply to (class)** field, enter the name of the class that you created in steps 29 to 32.

> **For example:**
>
> Enter `Embed-NotificationChannel-SMS`.

36. Click **Create and open**.
37. On the **Definition** tab of the Data Transform form, in the **Source** column for the *pyChannelName* property, provide the name of your Channel.

> **For example:**
>
> Enter `SMS`.

38. Set the *pxChannelRank* property value to a numeric value, to reflect the position of the Channel on the Notification Rule form.

©2025 Pegasystems Inc

> **NOTE:** The numbering of the list starts from the Channel following the `Web Gadget` Channel for Portal notifications, which is a default Channel that is not removable.

39. Click **Save**.

Result:

The following figure shows a sample configuration of the Data Transform:

Sample configuration of the Data Transform

Overriding the default *pyChannelConfiguration* Section:

40. In the navigation pane of Dev Studio, click **Records**.
41. Expand the **User Interface** category, and then click **Section**.
42. In the list of Section records, search for and open the *pyChannelConfiguration* Section from the *Embed-NotificationChannel* class.
43. On the **Design** tab of the Section configuration form, click the **View properties** icon next to the **pyChannelName** field:

©2025 Pegasystems Inc

44. In the **Value** field , enter the Channel name, and then click **Submit**.

> **For example:**
>
> Enter SMS.

45. Click **View properties** next to the **Enable Channel** checkbox, and then modify the value in the **Checkbox caption** field to include the new Channel name, and then click **Submit**.

> **For example:**
>
> Enter `Enable SMS notifications`.

46. In the editor menu, click **Data capture** > **Text input** to add a new Text field to the Section:

47. In the **Property** field, define the property that you want to associate with the Text field to provide the notification message.

©2025 Pegasystems Inc

For example:

Create a new property by clicking the crosshairs icon next to the **Property** field, and then configuring a new *.SMS* property. For more information, see Properties and Configuring a Text input control.

48. In the **Label value** field, enter `Message`, and then click **Submit**.

Result:

The following figure shows the new Section configuration:

49. Click **Save as**.
50. In the **Label** field, enter `pyChannelConfiguration`.
51. In the **Apply to (class)** field enter the name of the class that you created in steps 29 to 32.

For example:

Enter `Embed-NotificationChannel-SMS`.

52. Click **Create and open**.
53. **Optional:** To provide validation for the input in the **Message** field, override the *Validate* Activity by defining your own conditions and saving the Activity in the class that you created in steps 29 to 32.

©2025 Pegasystems Inc

For example:

The following Validate Activity in the *Embed-NotificationChannel-SMS* uses the *Property-Set-Messages* method to check if the **Message** field, which is sourced from the *.SMS* property, is not empty:

Validate Activity that checks for empty messages

For more information, see Creating an activity and Property-Set-Messages method.

Refreshing the *D_pzGetListOfChannels* Data Page, which holds a list of all Channels configured in the system:

54. In the navigation pane of Dev Studio, click **Records**.
55. Expand the **Data Model** category, and then click **Data Page**.
56. In the list of Data Pages, click the *D_pzGetListOfChannels* Data Page.
57. In the upper-right corner of the Data Page form, click **Actions** > **Run**.
58. In the **Value** field for the *DerivesFrom* parameter, enter `Pega-Notification`.
59. Select the **Flush all instances of this Data Page before execution** checkbox to delete any existing instances of the selected Data Page.
60. In the upper-right corner, click **Run**.

©2025 Pegasystems Inc

Result:

The new Channel appears on the **Channels** tab of the Notification Rule that you use to send notifications in your application:

©2025 Pegasystems Inc

Notification: Case status details [Available]

Channels

Channel configurations

Web gadget

Enable web gadget

Display section *

pyShowNotificationDefault

Default preference

Enable

Mobile

Enable push notification

Message

pyCaseDetails

Parameters

CaseID	.pyID
Stage	.pxCurrentStageLabel
Status	.pyStatusWork

Default preference

Enable

Email

Enable email

Subject *

pyCaseDetails

Parameters

CaseID	.pyID
Stage	.pxCurrentStageLabel
Status	.pyStatusWork

Correspondence *

pyCaseDetails

Default preference

Instant

SMS

Enable SMS notification

Message

©2025 Pegasystems Inc

Newly created Channel on the Notification Rule form

Related concepts

- Sections

Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools

Use Pega GenAI and collaborative tools to accelerate Case processing by empowering Case workers with AI-driven insights and dynamic features. These tools enable Case workers to adapt quickly to changing situations, collaborate effectively, and take the relevant steps to ensure efficient and accurate Case resolution.

By enriching your Case Types with Pega GenAI tools, optional Actions, and supporting features such as Pulse, you ensure that the Case workers who use your application, for example customer service representatives, have enough data and the right tools to quickly and efficiently process their Cases. As a result, you help your customers achieve their objectives and successfully resolve their Cases, even when unexpected measures are required.

- **Pega GenAI Coach in Case Management**
- **Responding to business exceptions in a workflow**
- **Configuring and working with optional Actions in Cases**
- **Preparing for collaboration with users by using Pulse**

Pega GenAI Coach in Case Management

Pega GenAI Coach is a powerful feature that is designed to provide intelligent guidance and assistance to users within Pega Platform™. At run time, Coach leverages generative

©2025 Pegasystems Inc

AI technology to offer personalized recommendations, step-by-step instructions, and best practices, assisting users in navigating applications and processes built on the Pega Platform.

Using Pega GenAI Coach

Pega GenAI Coach is a run-time feature within Pega Platform that serves as an intelligent assistant for users, providing personalized guidance and support. Users interact with Coach to receive context-aware recommendations, step-by-step guidance, and automation for navigating complex workflows and tasks within the application. Users can ask predefined questions related to an application and its processes and receive intuitive responses without requiring deep technical knowledge.

The figure below shows an example of a Coach that is accessible on an application landing page. The user selects predefined questions to help them develop their understanding of the Case, and the Coach responds with Case-specific data. The Coach replies with the information that is required for the user to progress with the application.

Configuring Pega GenAI Coach

Developers can configure Pega GenAI Coach to create specific Coaches with customized definitions and data sources based on their business requirements. This involves configuring the Coach starter question, quick-select questions, and the data sources that the Coach reads from to provide relevant guidance to users, as shown below:

©2025 Pegasystems Inc

Coach *configuration for definitions and data sources*

For more information, see Setting definitions for Pega GenAI Coach and Adding a data source to a Pega GenAI Coach.

Developers also have the flexibility to incorporate multiple Coaches within their applications. This can be particularly useful when different sections or functionalities of an application require specific Coach assistance. By having multiple Coaches, developers can tailor the guidance and recommendations that is provided to users based on the specific context or task at hand.

©2025 Pegasystems Inc

Switching between Coaches *at run time*

For more information, see Enabling Pega GenAI Coach in a Case Type or Data Object.

By making use of these configuration options, developers can ensure that Pega GenAI Coach aligns with their application's unique needs and provides tailored guidance to their users.

- **Enabling Pega GenAI Coach in a Case Type or Data Object**
- **Creating a new Pega GenAI Coach**

Related tasks

- Enabling Pega GenAI Coach in a Case Type or Data Object

©2025 Pegasystems Inc

- Creating a new Pega GenAI Coach
- Setting definitions for Pega GenAI Coach
- Adding a data source to a Pega GenAI Coach
- Adding a Pega GenAI Coach as a widget

Enabling Pega GenAI Coach in a Case Type or Data Object

Add a Pega GenAI Coach™ to a Case Type or data object in Constellation applications to help users make use of personalized generative AI guidance and assistance at run time, increasing both efficiency and scalability.

Pega GenAI Coach serves as a conversational assistant and mentor, providing Task summaries and guidance on implementing best practices to improve efficiency and accuracy. Coach can be customized for different Personas and roles, and enables users to ask questions with generative AI providing answers based on real-time Case data and pre-configured best practices.

While Coaches can be easily configured to ensure that each Coach is perfectly tailored to an organization's objectives and their employees' specific needs, Pega Platform™ also includes a default Pega GenAI Coach out of the box.

> **Before you begin:**
>
> Enable and configure generative AI in your application settings. For more information, see Connecting to the generative AI provider in Pega Cloud.

Adding a Pega GenAI Coach to a full page View

1. Open the Case Type or Data Object where you want to enable Pega GenAI Coach:

©2025 Pegasystems Inc

Choices	Actions
Enable Coach in a Case Type	In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
Enable Coach in a data object	In the navigation pane of App Studio, click **Data**, and then click the data object that you want to open.

> **NOTE:** You can also enable Pega GenAI Coach on landing pages. For more information, see Creating a landing page for an application.

2. Click the **UX** tab.
3. On the **Full Page View** tab, in the **GenAI™ Coach** section, select the **Enable GenAI™ Coach in Utilities** checkbox.
4. In the **Coaches** section, click **Add**.
5. Select a Coach from the list of available Coaches, then click **Add**.

> **NOTE:** **GenAI Coach Default** is an out-of-the-box Case Type Coach that is available to you after you enable Pega GenAI™. For information on creating new Coaches, see Creating a new Pega GenAI Coach.

6. Click **Add**.
7. **Optional:** To add more Pega GenAI Coaches, click **Add** and then repeat steps 5 and 6.
   At run time, users can choose which Coach to use by clicking the name of the Coach and selecting it in a list:

©2025 Pegasystems Inc

Switch between Coaches at run time

8. Click **Save**.

   The following video shows how to enable a Pega GenAI Coach for a Case Type:

Result:

The Pega GenAI Coach is added to the list of Coaches in your Case Type and is displayed in the utilities pane.

The following figure shows a custom Pega GenAI Coach, entitled Loan Coach, in the utilities pane at run time:

©2025 Pegasystems Inc

GenAI Coach in the utilities pane

What to do next:

Create a new and customized Pega GenAI Coach like the one in the figure above. By creating multiple customized Coaches, you increase the scope of what Pega GenAI Coach can assist with at run time. For more information, see Creating a new Pega GenAI Coach.

Related tasks

- Creating a new Pega GenAI Coach
- Setting definitions for Pega GenAI Coach

©2025 Pegasystems Inc

Creating a new Pega GenAI Coach

Create a new Pega GenAI Coach™ to run user-defined generative AI procedures at run time. By creating multiple custom Pega GenAI Coaches, you increase the scope of what Pega GenAI Coach can assist with, and improve efficiency across your application.

For example, a company wants to enhance the effectiveness of sales representatives when closing sales by using Pega GenAI Coach to help them create a well-formatted close plan. The close plan is based on recent activity data and Pulse posts in an Opportunity Management Case Type. The default Pega GenAI Coach does not meet their business needs, so they create a new Coach with a custom configuration.

> **Before you begin:**
>
> Enable and configure generative AI in your application settings. For more information, see Connecting to the generative AI provider in Pega Cloud.

Creating a Pega GenAI Coach

1. In the header of Dev Studio, click **Create** > **Generative AI** > **GenAI Coach**.
2. In the **Label** field, enter a name for the Coach.

   > **For example:**
   >
   > Enter `Close Plan Coach`.

3. In the **Apply to** field, select the class to which you want to add the Coach.

   > **For example:**
   >
   > For a Case Type entitled Opportunity Management, select **ABCEDEF-SalesAut_1-Work-OpportunityManagement**.

4. In the **Add to ruleset** field, select the Ruleset to which you want to add the Coach.
5. Click **Create and open**.

©2025 Pegasystems Inc

6. Click Save.

What to do next:

Set the definitions of the new Coach to define its role. For more information, see Setting definitions for Pega GenAI Coach.

Add data sources to the new Coach to increase the scope of accessible data. For more information, see Adding a data source to a Pega GenAI Coach.

- **Setting definitions for Pega GenAI Coach**
- **Adding a data source to a Pega GenAI Coach**
- **Selecting a generative AI model for a Coach**
- **Adding a Pega GenAI Coach as a widget**

Related tasks

- Setting definitions for Pega GenAI Coach

Setting definitions for Pega GenAI Coach

Set instructions for Pega GenAI Coach™ to define its role and prompt responses. By entering clear and detailed instructions for a Coach, you help ensure that it operates effectively, responsibly, and in alignment with business goals.

Before you begin:

- Enable and configure generative AI in your application settings. For more information, see Connecting to the generative AI provider in Pega Cloud.

PEGA ©2025 Pegasystems Inc

- Add a Pega GenAI Coach to your Case Type. For more information, see Enabling Pega GenAI Coach in a Case Type or Data Object or Creating a new Pega GenAI Coach.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Generative AI** category, and then click **GenAI Coach**.
3. On the **GenAI Coach** tab, click the Coach for which you want to edit the definition .
4. In the **System Instructions** field, enter a definition that summarizes the role that you want the Pega GenAI Coach to assume.

   For example:

   `You are a Pega empowered case manager AI assistant that helps people find information and navigate a Case.`

Adding a starter question to a Coach

5. In the **Starter Question** section, in the **Question** field, enter an initial question to ask Coach when you click **Start chat** at run time.

   For example:

   Enter `What are the next steps in this loan application?`. The following figure shows a Coach starter question at run time:

©2025 Pegasystems Inc

A Coach starter question at run time

6. **Optional:** To enter further instructions for Coach to consider, to provide a more relevant and precise response to the starter question. in the **Starter Question** section, in the **Additional Instructions** field, enter some relevant text.

> **NOTE:** **Additional Instructions** are not displayed in the Coach conversation.

Adding quick-select questions to a Coach

©2025 Pegasystems Inc

7. In the **Quick-Select Question** section, in the Question field, enter a question to ask Coach when you select it from the Ask list at run time.

For example:

Enter `What personal information do I need to progress my application?`, `How would having many dependents affect the outcome of my application?` and `What will happen after I submit my information?`. The following figure shows a Coach with quick-select questions in the Ask Loan Coach list at run time:

©2025 Pegasystems Inc

A Coach with quick-select question suggestions at run time

8. **Optional:** To have the Coach consider further instructions, to provide a more relevant and precise response to the quick select question, in the **Quick-Select Question** section, in the Additional Instructions field, enter some relevant text.

> **NOTE:** Additional Instructions are not displayed in the Coach conversation.

9. **Optional:** To add more quick-select questions, click Add.
10. Click Save.

What to do next:

Add data sources to the new Coach to increase the scope of accessible data. For more information, see Adding a data source to a Pega GenAI Coach.

Related tasks

- Enabling Pega GenAI Coach in a Case Type or Data Object
- Creating a new Pega GenAI Coach

Adding a data source to a Pega GenAI Coach

Add a data source to a Pega GenAI Coach™ to increase the scope of accessible data. As a result, you can improve the accuracy of Coach responses and expand the feature's functionality.

©2025 Pegasystems Inc

Before you begin:

- Enable and configure generative AI in your application settings. For more information, see Connecting to the generative AI provider in Pega Cloud.
- Add a Pega GenAI Coach to your Case Type. For more information, see Enabling Pega GenAI Coach in a Case Type or Data Object and Creating a new Pega GenAI Coach.
- Create Data Pages for the data that you want your Coach to source information from. For more information, see Creating a Data Page.

NOTE: As a best practice, add at least one data source to increase the accuracy of the Pega GenAI Coach. Although Pega GenAI Coach can work without any data sources, providing them highly increases the accuracy of its responses.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Generative AI** category, and then click **GenAI Coach**.
3. On the **GenAI Coach** tab, click the Coach for which you want to edit the definition .
4. Click the **Data sources** tab.
5. Click **Add Data Page**.
6. In the **Name** field, enter a name for the data source.
7. In the **Data Page** field, enter the name of the Data Page from which you want to source data.

   For example:

   Enter `D_ClientMeetings`.

8. In the **Parameters** field, enter the required parameters for the Data Page.

©2025 Pegasystems Inc

> **NOTE:** Required parameters vary depending on the Data Page that you select.

9. In the **Apply Data Transform** field, enter a Data Transform to apply to the Data Page.

> **NOTE:** You require a Data Transform to map eligible fields. If you do not specify a Data Transform, the system converts the entire Data Page to JSON format. If the Data Page is a List type, create the Data Transform in the *Code-Pega-List* class. Alternatively, create the Data Transform in the *Data Page* Object Type.

10. Click **Save**.

For example:

The following figure shows an example of three data sources added to a Close Plan Coach. The data sources include:

- A *Pulse summary* data source for summarizing Pulse conversations in a Case.
- A *Client meetings summary* data source for collecting information to summarize meeting contents between a sales representative and a client.
- A *Client emails summary* data source for collecting information to summarize email correspondence between a sales representative and a client.

©2025 Pegasystems Inc

Adding data sources to a Coach

Related tasks

©2025 Pegasystems Inc

- Setting definitions for Pega GenAI Coach
- Creating a new Pega GenAI Coach
- Enabling Pega GenAI Coach in a Case Type or Data Object

Selecting a generative AI model for a Coach

Ensure that Pega GenAI Coach™ produces answers that best fit your business circumstances by selecting a suitable generative AI model. You select an AI model depending on your business use case. Some models are designed for fast and efficient processing, while other models perform better at analytical and complex tasks.

> **Before you begin:**
>
> Enable and configure generative AI in your application settings. For more information, see Connecting to the generative AI provider in Pega Cloud.

Selecting the AI model for Coach is available from Pega Platform™ release 24.2.2.

The selection of the available models depends on your could provider and your region. For more information, see Deployment regions for Pega Cloud.

> **NOTE:** If your environment is hosted on Pega Cloud® for Government, you can only select models compliant with Federal Risk and Authorization Management Program (FedRAMP) regulations, which are currently AWS models.

Pega Platform release 24.2.2 includes two models by default:

Pega-Default-Fast

Offers faster processing by providing less detailed responses. This model is more lightweight and is by default mapped to the relevant model available for your cloud provisioning.

©2025 Pegasystems Inc

Pega-Default-Smart

Provides more complex responses but might require more processing. This model is more robust and is by default mapped to the relevant model available for your cloud provisioning.

If your cloud provider releases a newer version of the model, Pega Platform automatically includes the updated versions for the models that you selected as *Pega-Default-Fast* and *Pega-Default-Smart*, saving you time on manual adjustments.

> **NOTE:** After the model update, ensure that you test Pega GenAI™ features that use the updated model.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Generative AI** category, and then click **GenAI Coach**.
3. In the list of Coach instances, select the Coach that you want to edit.
4. On the **Advanced** tab, in the **GenAI model** list, select the model that you want to use.
5. Click **Save**.

Adding a Pega GenAI Coach as a widget

Enhance the accessibility and positioning of a Pega GenAI Coach™ by adding it as a widget within a tab. This increases the overall user experience by enabling developers to provide targeted Coach guidance to users, resulting in improved productivity.

> **Before you begin:**
>
> - Enable and configure generative AI in your application settings. For more information, see Connecting to the generative AI provider in Pega Cloud.
> - Create a Pega GenAI Coach with custom definitions. For more information, see Creating a new Pega GenAI Coach and Setting definitions for Pega GenAI Coach.

©2025 Pegasystems Inc

> **IMPORTANT:** Widgets created in Pega Platform™ before version '24.2 must be recreated to benefit from the '24.2 Coach enhancements.

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 **UX** tab.
3. On the **Full Page View** tab, in the **Tabs** section, click **Add**.
4. In the **Name** field, enter a suitable name for the tab, and then click **Submit**.

   > **For example:**
   >
   > Enter the name `Pega GenAI Coach`.

5. Click the newly created tab.

A newly created Close Plan tab

6. In the **Edit tab** pane, in the **Region A** section, click **Add**.
7. Click **Widgets**, select **GenAI Coach**, and then click **Add**.
8. Click **Save**.

©2025 Pegasystems Inc

The following video shows a user adding a default Pega GenAI Coach to Case Type settings, and then adding Coach as a widget to a full page View:

Responding to business exceptions in a workflow

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 Process, 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 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 **Lifecycle**.
   b. Click a Process in a Stage and then click **Open Process**.
   c. On the **Diagram** tab of the 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 Process.
   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.

©2025 Pegasystems Inc

h. Click **Submit**.
i. Click **Save**.

5. Activate the ticket in a flow that detects your business exception.

- To automatically activate the ticket:

  - Return to the **Workflow** tab of your Case Type and click **Lifecycle**.
  - Click a Process in a Stage and then click **Open Process**.

    You can set a ticket in one Process and raise it in another as long as both Processes are in the same Case.

  - On the **Diagram** tab of the form, add the Utility shape to the Process diagram where the exception occurs.
  - Double-click the Utility shape and then configure it to call the *@baseclass.SetTicket* activity with the name of your ticket as an input parameter.
  - Click **Submit**.
  - Click **Save**.
- To manually activate the ticket:
  - Creating Flow Actions.

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

  - Return to the **Workflow** tab of your Case Type and click **Lifecycle**.
  - Click a Process in a Stage and then click **Open Process**.

    You can set a ticket in one Process and raise it in another as long as both Processes are in the same Case.

  - On the **Diagram** tab of the form, double-click an Assignment shape to open the **Assignment properties** dialog box.

©2025 Pegasystems Inc

> **TIP:** To make your Flow Action available for more than one Assignment, associate it with a Stage in the Lifecycle of the Case Type or the Case Type itself.

- In the **Advanced** section, click **Local Actions**.
- Click **+ Add local Action**.
- In the **Local Action** field, press the Down Arrow key and select the name of your Flow Action.
- Click **Submit**.
- Click **Save**.

6. **Optional:** To change the limits that prevent infinite loops in your Processes, 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.

- **Creating a ticket**
- **Policy overrides and suspended work items**
- Changing the path of a Process
- Adding optional Actions to Cases
- Adding optional Actions to Stages

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**.

©2025 Pegasystems Inc

2. On the **Create** form, enter values in the fields to define the context of the ticket.

   Options include:

   a. In the **Label** field, enter text that describes the purpose of the ticket.

   b. In the **Apply to** field, press the Down Arrow key and select the class that defines the Scope of the ticket.

      The class controls which Rules the ticket can use. It also controls which Rules can call the ticket.

   c. In the **Add to Ruleset** field, select the name and unlocked version of a Ruleset that stores the ticket.

   d. To change the default identifier for the ticket, click **Edit**, and then provide a unique value in the **Identifier** field.

3. Click **Create and open** to open the Ticket form.

4. Click **Save**.

You can make your ticket functional by setting and activating it in different Processes in the Lifecycle of a Case.

- Responding to business exceptions in a workflow
- Copying a Rule or data instance
- Creating a Rule specialized by circumstance

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.

©2025 Pegasystems Inc

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.

PEGA ©2025 Pegasystems Inc

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

©2025 Pegasystems Inc

- 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.

Related information

- How to implement business exception processing with policy overrides

Configuring and working with optional Actions in Cases

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 an Assignment only when the Assignment is relevant to the Case.

Optional Actions may also be common Case processing Actions that may be invoked at varying times during the Case. For example, Pega Platform™ '24.2 provides the *Change Stage* optional Action that may be called at any time during Case processing.

©2025 Pegasystems Inc

Depending on the business need, the optional Action can be a Process if a series of Steps are required or a user Action if exactly one Step is required. Optional Actions are accessible at run time from the **Actions** menu in the Case.

Use an optional Action to configure a single Action that users can invoke at any time during a particular Stage, or at any time during the Case, such as updating information in a single screen or transferring the Case to another user.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

- **Adding optional Actions to Cases**
- **Adding optional Actions to Stages**
- **Displaying optional Actions conditionally**
- **Adding supporting optional Processes to Cases**
- **Adding supporting optional Processes to Stages**
- **Displaying supporting Processes Conditionally**
- Configuring promoted Actions on a Full Case View
- Automating work by creating Case Types
- Adding optional Actions to paused Processes
- Working with optional Actions in data objects

PEGA ©2025 Pegasystems Inc

Adding optional Actions to Cases

Allow customer service representatives (CSRs) to perform out-of-sequence Assignments that do not influence the main events in the Case Lifecycle 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 Assignment to update the phone number without moving the Auto Loan Case to a different Step or Stage.

For relevant training materials, see the Adding optional Actions to a workflow module on Pega Academy.

1. In the navigation pane of 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 an Assignment, click **Collect information**.
   - To add a Process, click **Process**, and then add Steps by clicking **Step**.

     For more information about configuring Steps, see Steps in a Case Lifecycle.

   - To add a Multi-step Form, click **Multi-step 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 **User action** property pane, in the **Visible** field, define the visibility of the Action.
6. Save your changes by clicking **Save** or **Save and run**.

©2025 Pegasystems Inc

Result:

At run time, optional Actions appear in the header of a Case.

What to do next:

Give users quick access to frequently used Actions. For more information, see Configuring promoted Actions on a Full Case View.

- Case Lifecycle elements
- Adding optional Actions to Stages
- Displaying optional Actions conditionally

Adding optional Actions to Stages

Provide CSRs with an ability to perform out-of-sequence Assignments 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 the Candidate details Stage that includes the Collect personal details, Collect education history, and Collect working experience Processes. If a job candidate gives a CSR a new phone number, the CSR can update the number at any point when processing the Stage.

For relevant training materials, see the Adding optional Actions to a workflow module on Pega Academy.

1. In the navigation pane of 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:

©2025 Pegasystems Inc

- To add an Assignment, click **Collect information**.
- To add a Process, click **Process**, and then add Steps by clicking **Step**.

  For more information about configuring Steps, see Adding Steps to Processes.

- To add a Multi-step Form, click **Multi-step 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 **User action** property pane, in the **Visible** field, define the visibility of the Action.
6. Save your changes by clicking **Save** or **Save and run**.

Result:

At run time, the Action appears in the Actions menu in the header of a Case when the Case enters the Stage.

What to do next:

Give users quick access to frequently used Actions. For more information, see Configuring promoted Actions on a Full Case View.

- Case Lifecycle elements
- 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 Assignments only when the Assignments are relevant to the Case that CSRs currently process. By defining conditions for displaying optional Assignments, you accelerate Case resolution and save time because users

©2025 Pegasystems Inc

work only on significant Assignments. Additionally, you can make optional Actions available after the Case is resolved. For example, you can help a user correct a typing mistake in an already resolved request.

> **Before you begin:**
>
> Add an optional Action to your Case Lifecycle. 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 Lifecycle** 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 Action 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.

©2025 Pegasystems Inc

Choices	Actions
	For example: `Case status is equal to New` d. **Optional:** To define more Conditions, click the **Add row** icon, and then repeat step 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. **Optional:** To make the Action available for resolved Cases as well, select the **Allow When Case is resolved** checkbox.

©2025 Pegasystems Inc

> **NOTE:** You can make the Action visible exclusively for resolved Cases by adding a Condition, such as `Is resolved`.

6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **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 optional Processes to Cases

Complement your Case Lifecycle with a supporting Process to allow Case workers, for example customer service representatives, to determine when a Case requires additional processing.

A supporting Process runs independently of the Case Lifecycle, 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.

PEGA ©2025 Pegasystems Inc

> **NOTE:** Supporting Processes, also known as optional Processes, are optional Actions that require a series of Steps to be completed.

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 Lifecycle** section, click **Optional Actions**.
3. In the **Case-wide Actions** section, click **Add Action**.
4. In the Action menu, define 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 selects it manually from the **Actions** menu of the Case, in the **Process** property pane, select the **Manually start** checkbox, and then define the Condition for displaying the supporting Process. For more information see Displaying supporting Processes Conditionally.
6. **Optional:** To start the Process when a user creates a Case, in the **Process** property pane, select the **Automatically start when this Case starts** checkbox.
7. **Optional:** If you build a Constellation application, to enable users to return to the previous Step in Case Processing, in the **Process** property pane, select the **Allow users to go back to the previous Step** checkbox.
8. **Optional:** To enable Case workers to dismiss a supporting Process, enable an out-of-the-box Action that removes the started Process:
   a. In the header of the supporting Process, click **Configure Process**.
   b. On the toolbar, click **Open Process in Dev Studio**.
   c. Open the property dialog box by double-clicking the Assignment shape.
   d. In the **Advanced** section, expand the **Local Actions** category, and then in the **Local Action** field, enter `pxDismiss`.
   e. Click **Submit** > **Save**.
9. Save your changes by clicking **Save** or **Save and run**.

©2025 Pegasystems Inc

Result:

At run time, you can see the Process in the **Actions** menu in the Case header at any point of the Case Lifecycle. When you select the Process, 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 Steps to Processes.
- Define when the user sees the Process by configuring Visibility Conditions. For more information, see Displaying supporting Processes Conditionally.

Give users quick access to frequently used Actions. For more information, see Configuring promoted Actions on a Full Case View.

- Adding supporting optional Processes to Stages
- Displaying supporting Processes Conditionally

Adding supporting optional Processes to Stages

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 CSR 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 the Collect accident details Stage.

PEGA ©2025 Pegasystems Inc

> **NOTE:** Supporting Processes, also known as Optional Processes, are optional Actions that require a series of Steps to be completed.

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 Lifecycle** section, click **Optional Actions**.
3. In the **Stage-only Actions** section, click **Add Action** below the Stage to which you want to add a supporting Process.
4. Define a supporting Process:
   - To add a new Process, click **Process**, and then replace the default label for the Process with a name that describes the purpose.
   - To add an existing Process, click **More** > **Processes** > *[Process name]*, and then click **Select**.
5. **Optional:** To define the Conditions for displaying the supporting Process, perform step 4 in Displaying supporting Processes Conditionally.
6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **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.

©2025 Pegasystems Inc

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 Steps to Processes.
- Define the Conditions for when users see the Process. For more information, see Displaying supporting Processes Conditionally.

Give users quick access to frequently used Actions. For more information, see Configuring promoted Actions on a Full Case View.

- Processes in a Case Lifecycle
- Adding optional Actions to Stages
- Steps in a Case Lifecycle

Displaying supporting Processes Conditionally

Save time during Case processing by ensuring 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 Lifecycle** section, click **Optional Actions**.
3. Click a supporting Process in 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 of the Case Lifecycle. If you select the Process in the **Stage-only**

©2025 Pegasystems Inc

Actions section, CSRs can process the Action when a Case enters a particular Stage.

4. In the **Process** properties pane, 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 step 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.

©2025 Pegasystems Inc

Choices	Actions
	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.

> **NOTE:** If you are an advanced developer, you might want to define the Visibility Conditions with a When Rule or an Expression. These options are available only in Dev Studio, which you access from the header of your studio by clicking **Switch Studio** > Dev Studio.
>
> For more information about building Expressions, see Building Expressions with the Expression Builder.

5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

Result:

At run time, when a Case reaches a supporting Process that meets the Conditions, the user sees the Process.

- Adding supporting optional Processes to Cases
- Adding supporting optional Processes to Stages

Preparing for collaboration with users by using Pulse

Improve collaboration and conversation among users within a specific context, such as a Case, by adding the Pulse collaboration gadget to your application. By sharing information in Pulse, users can work together to resolve their work more quickly.

In Pulse, users can post, view, and reply to messages, to promptly exchange information. To enrich messages, users can attach content to posts and style the text. For example, users can bold important information within the message. Ensure the best-tailored experience when using Pulse by defining default feed sources, so that users immediately see only the relevant information.

By configuring a message display, you define what elements a message shows, so that users find all the vital information in the Pulse posts. For instance, you can define a message title to include a link to a Case.

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

You can use Pulse to collaborate in end-user Portals when you work on a Case, update your user profile, or view your activity feed. Because Pulse is a reusable tool, you can

©2025 Pegasystems Inc

embed it in other places in your application. For improved communication in a Case context, you can configure Pulse for different Case Types, so that users collaborate to reach a Case resolution faster. You configure Pulse for a Case Type directly in Case Designer.

To post and view messages in Pulse, you embed the *pxFeed* gadget in your application. The *pxFeed* gadget shows message replies, messages in Cases that you follow, messages that reference users, profile messages, bookmarked messages, and application messages. Include this gadget within a Section or harness in your application to view the activity feed, for example, in the dashboard of the Case Manager Portal. You can use the gadget to display the feed within a specific context, such as a Case, a team, or a custom context.
You can also configure the Pulse gadget so that users can post messages directly to the activity feed, or to show posts from internal or external data sources, such as the following Data Page APIs:

D_pxPostRepliesFeed
: Retrieves replies to user posts

D_pxFollowedCasesFeed
: Retrieves posts to Cases followed by users

D_pxMentionsFeed
: Retrieves posts that reference users

D_pxProfileFeed
: Retrieves posts to user profile pages

D_pxBookmarkFeed
: Retrieves posts that users bookmark

D_pxPosts
: Retrieves posts for a Case, Rule, custom or operator context.

Examples of collaboration in Pulse

The following scenarios are examples of how you can use Pulse to increase collaboration:

PEGA ©2025 Pegasystems Inc

- Customer service representatives (CSRs) and product managers that resolve customer issues
- Account executives that track a sales opportunity
- Marketing personnel that discuss a campaign design
- Financial services professionals that approve a loan or credit card request

Enable collaboration on Cases by completing the following tasks:

- **Adding the Pulse gadget to your application**
- **Enabling users to post messages in the activity feed**
- **Creating feed sources for activity feeds**
- **Adding feed sources to activity feeds**
- **Determining default feed sources**
- **Configuring the display of feed sources**
- **Configuring Pulse email notifications**
- **Configuring Pulse feed sources for Constellation**
- **Markdown in Pulse**

Related tasks

- Collaborating with users by using Pulse
- Sourcing attachments from external storage

Adding the Pulse gadget to your application

Provide a collaboration tool for users of your application by adding the Pulse gadget to your application. By posting, viewing, and commenting on messages, users can work together to resolve Cases faster. Users that post Pulse messages can make their posts more meaningful by adding attachments, including links, or by formatting the text.

PEGA ©2025 Pegasystems Inc

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **User Interface** category, and then click **Section**.
3. Open a Section in which you want to add a Pulse gadget.
4. Add the Pulse gadget:
   - If you edit the Section in the View editor, click **Add** > **Data display** > **Embedded Section**.
   - If you use a full Section editor, click **Structural**, and then drag **Embedded Section** to a place in the layout where you want to display Pulse.
5. In the **Section include modal** window, in the **Section** field, enter pxFeed.
6. Click **Submit**.
7. Click **Save**.

> **What to do next:**
>
> Create feed sources that you can add to activity feeds. For more information, see Creating feed sources for activity feeds.

- Collaborating with users by using Pulse
- Collaboration with users by using Spaces

Enabling users to post messages in the activity feed

Enable users to post messages in an activity feed for a specific context. For example, a user can make release announcements in the context of an application for all the application users to see.

Users can view their activity feeds in the Case Manager and Case Worker Portals when they click the **Pulse** menu option.

> **NOTE:** This procedure applies to traditional UI Pega Platform applications. For applications based on Constellation, see Configuring Pulse feed sources for Constellation.

©2025 Pegasystems Inc

1. In Dev Studio, search for the *pyPulseWrapper* Section.
2. Override the *pyPulseWrapper* Section that includes the *pxFeed* Section.
3. On the **Design** tab of a Section or harness, click the *pxFeed* Section.
4. Open the **Layout Properties** pane by clicking the **Settings** icon.
5. On the **Parameters** tab, in the **Configure post settings** section, select the **Enable post** checkbox.
6. In the **Context key** field, define the context for posting messages by entering a key value.

   > **For example:**
   >
   > Enter: `Application.pxInsName`

7. In the Section, select the message types that users can post:
   - To allow users to post public messages, select **Message**.
   - To allow users to post private messages, select **Private post**.
   - To allow users to create and assign tasks, select **Task**.

   > **Result:**
   >
   > The post types are displayed in a menu when you open Pulse and click **Post**. For more information, see Posting a message in Pulse.

8. **Optional:** Update the description of a selected message type:
   a. Next to the message type, click the **Edit** icon.
   b. In the **Feed name** field, update the description.
   c. Click **Done**.
9. Click **Submit**.
10. Click **Save**.

- Collaborating with users by using Pulse
- Creating an embedded Section
- Creating feed sources for activity feeds
- Adding the Pulse gadget to your application

PEGA ©2025 Pegasystems Inc

Creating feed sources for activity feeds

Inform users about events that are relevant to them by creating feed sources for messages from internal or external data sources. After you create a feed source, you can include it in a user activity feed.

For example, you can configure a Pulse feed to display posts from members of different departments.

> **NOTE:** This procedure applies to traditional UI Pega Platform applications. For applications based on Constellation, see Configuring Pulse feed sources for Constellation.

> **Before you begin:**
>
> To post and display messages, add the Pulse gadget to your application. For more information, see Adding the Pulse gadget to your application.

1. In the header of Dev Studio, click **Create menu** > **Process** > **Pulse Feed**.
2. In the **Pulse feed record configuration** section, in the **Label** field, enter a short description for the feed source that you want to create.

   > **For example:**
   >
   > Describe messages that users post for a team by entering `TeamPosts`.

3. In the **Apply to** field, select a class to which this feed source applies.
4. In the **Add to Ruleset** field, select the name of the Ruleset that contains the feed source.
5. Click **Create and open**.
6. In the **Feed label** field, enter a description for the new feed.

©2025 Pegasystems Inc

By default, the system populates the feed label with the label that you provide in the **Label** field in step 2.

7. In the **Data page** field, enter the Data Page that retrieves the messages for the feed.
8. In the fields that are displayed based on the Data Page that you choose, configure additional parameters.

   For example:

   If you use the *D_pxPostRepliesFeed* Data Page to retrieve replies to user messages, in the **UserID** field, enter `pxRequestor.pyUserIdentifier`.The system uses this value to fetch replies to the messages that logged-in users post.

9. Click **Save**.

What to do next:

- Add a feed source to a user activity feed. For more information, see Adding feed sources to activity feeds.
- Personalize the feed items in your application by configuring the display settings. For more information see, Configuring the display of feed sources.
- Enable posting Pulse comments in Cases. For more information, see Configuring Pulse for Case Types.
- To make the feed source available for reuse in App Studio, mark it as relevant. For more information, see Marking a record as relevant.

- Collaborating with users by using Pulse
- Creating an embedded Section
- Enabling users to post messages in the activity feed
- Adding the Pulse gadget to your application

©2025 Pegasystems Inc

Adding feed sources to activity feeds

Ensure that users can view relevant messages by adding a feed source to their activity feed. For example, you can add an external feed source, so that customer service representatives (CSRs) that work with finances can see posts about currency rates.

> **NOTE:** This procedure applies to traditional UI Pega Platform applications. For applications based on Constellation, see Configuring Pulse feed sources for Constellation.

Before you begin:

- Add the Pulse gadget to your application. For more information, see Adding the Pulse gadget to your application.
- Create a feed source. For more information, see Creating feed sources for activity feeds.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **User Interface** category, and then click **Section**.
3. Open a Section that contains the Pulse gadget.
4. In the *pxFeed* Section, click **Edit this Section**.
5. In the **Cell Properties** window, click the **Parameters** tab.
6. In the **Configure additional feed sources** section, click **Add feed source**, and then specify the feed source:
   a. In the **Name** field, enter the name of the feed source that you want to add.
   b. **Optional:** To display messages from the feed source every time a user opens the Pulse gadget, select the **Display feed by default** checkbox.
   c. Click **OK**.
7. Click **Submit**.
8. Click **OK**.

©2025 Pegasystems Inc

- Collaborating with users by using Pulse
- Configuring Pulse for Case Types
- Configuring the display of feed sources
- Posting a message in Pulse

Determining default feed sources

Ensure that users see only relevant posts by specifying which feed sources Pulse displays by default.

> **NOTE:** This procedure applies to traditional UI Pega Platform applications. For applications based on Constellation, see Configuring Pulse feed sources for Constellation.

> **Before you begin:**
>
> Add a feed source for the activity feed. For more information, see Creating feed sources for activity feeds.

1. In the header of Dev Studio, search for the Section that contains the *pxFeed* Section that you want to configure, for example, *pyPulseWrapper*.
2. On the **Design** tab of the Section or harness, click the *pxFeed* Section.
3. Open the **Layout Properties** pane by clicking the **View properties** icon.
4. In the **Layout Properties**, click the **Parameters** tab.
5. Next to the feed source that you want to mark as the default source, select the **Default** checkbox.
6. Click **Submit**.

> **Result:**
>
> Users can see messages from default feed sources.

©2025 Pegasystems Inc

> Users can override the default settings at run time by filtering messages; however, the default changes are restored after reloading the page.

- Collaborating with users by using Pulse
- Enabling users to post messages in the activity feed

Configuring the display of feed sources

Configure the display of your Pulse feed sources to ensure that users see relevant information about a feed item, such as an icon or the profile image of the users related to the corresponding feed item.

For example, you can configure the messages that your application posts automatically to display your company logo and a Case ID as the title of the message.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Process** category, and then click **Pulse Feed**.
3. Open a Pulse feed that you want to edit.
4. In the **Pulse feed display settings**, in the **Icon** list, select the type of icon that you want to display in the message:

Choices	Actions
Display a user profile image	a. In the **Icon** list, select **User**. b. In the **User reference** field, enter a property that references the user whose profile image you want to display **For example:** To display the profile image of the user who posted the message, enter `.pyUser`.

©2025 Pegasystems Inc

Choices	Actions
Display an icon that a system provides	a. In the **Icon** list, select **Icon class**. b. Next to the **Icon class** field, click the **Open the icon class picker** icon. c. In the **Icon class picker** window, select an icon that you want to display next to the Pulse message.
Display an image from the Image Catalog	a. In the **Icon** list, select **Image**. b. Next to the **Image reference** field, click the **Show Image Viewer** icon. c. in the **Search** field, enter a key word or the name of the image that you want to use, and then click **Find**. **For example:** To view the different default logos that the system provides, enter Logo. d. Select the image. **NOTE:** Only traditional UI Pega Platform applications support adding images from the image catalog. Applications based on Constellation do not provide this option.

5. In the **Title** list, select a display format for the title of the message:

©2025 Pegasystems Inc

- To display the title as a value, in the **Title** list, select **Property**, and then in the **Property reference** field, enter a property that stores the value that you want to use.

  For example, to display a Case label as a title, enter `.pyLabel`.

- In a standard Pega Platformapplication, to display the title as a Section, in the **Title** list, select **Section**, and then in the **Section reference** field, enter a property that references the Section that you want to use.
- In an application based on Constellation, to use more complex text for the title, for example a sentence, in the **Title** list, select **Field value**, and then in the **Field value reference** field, enter a property that references the field value that you want to use.

6. In the **Time** field, enter the property that retrieves the time at which a user adds the message.

   For example:

   Enter *.pxUpdateDateTime*.

7. For a traditional UI Pega Platform application, in the **Message type** field, select a display format for the message:
   - To display a message as plain text, select **Message**, and then in the **Property reference**, enter a custom property.
   - To display a message as a Section, select **Section**, and then in the **Section reference**, enter a property that references a custom Section.
8. For an application based on Constellation, in the **Message reference** field, enter the name of a view that you want to use for displaying the messages.
   For more information about views in Constellation, see Working with Views.
9. **Optional:** To allow users to comment on a message, in the **Comments** section, select the **Allow comments** checkbox, and then, enter the custom property from the Data Page that uniquely identifies the feed item.

©2025 Pegasystems Inc

> **For example:**
>
> Enter `Application.pzInsKey`.

The **Allow comments** checkbox is available only for some Pega-delivered feed source types, such as *pxMySpacesFeed*, which displays Pulse messages from spaces in which a user is a member.

10. Click **Save**.

- Viewing your activity feed
- Posting a message in Pulse
- Collaborating with users by using Pulse

Configuring Pulse email notifications

Inform users in real-time about updates to their Pulse conversations by sending email notifications. In Pulse, you can choose to receive emails when another user references you, likes your messages, posts on your profile, posts comments on a conversation in which you participate, or posts a comment on a Case that you follow. As a result, you enhance communication and speed up Case resolution.

For example, when team members have a Pulse conversation about preparing a loan offer for a VIP customer, each member can receive email notifications about any activity in the conversation. To save time, users can also reply to Pulse messages by sending an email.

> **Before you begin:**
>
> Define how users want to receive notifications. For more information, see Setting notification preferences.

Receiving Pulse notifications by email helps users resolve cases more quickly, because users can exchange information directly from their email inboxes without logging in to

©2025 Pegasystems Inc

an application. In the example of preparing a loan offer, a user can receive email notifications when:

- Other users post Pulse messages in the Case.

  If the message includes an attachment, the user receives the attachment in the email.

- Other users reference the email recipient in a Pulse message.
- Other users like a message sent by the email recipient.

To resolve the Case more quickly, the user can respond to Pulse messages by replying directly to the Pulse notification from their inbox, without logging in to an application. Pulse then displays the response as a message in the conversation. To provide more context and information, the user can include an attachment in their email, that an application attaches to the Pulse message.

You can enable users to receive and respond to Pulse email notifications by performing the following actions:

- Integrating your application with an email provider
- Operators
- Collaborating with users by using Pulse

Configuring an email account for Pulse notifications

To notify users about any activity in their Pulse conversations by email, configure an email account to process Pulse notifications. As a result, users can immediately know about new Pulse messages in Cases that they follow, replies to their messages, and when other users reference them in Pulse comments. Immediate exchange of information can help users resolve work faster.

For example, if a customer service representative (CSR) posts a Pulse message with a question about the maximum loan amount that a customer can receive, and references

PEGA ©2025 Pegasystems Inc

a manager in the message, the manager receives an email that contains the message and can immediately respond to the question.

1. In the header of Dev Studio, click **Create** > **Integration-Resources** > **Email Account**.
2. In the **Short description** field, briefly describe the purpose of the email account.

   For example:

   Enter `Notify about Pulse`.

3. In the **Account Name** field, enter `PulseNotifications`.

   NOTE: You can have only one account with the name `PulseNotifications` in the system.

4. Click **Create and open**.
5. Configure the email account as appropriate.
   For more information, see Configuring outbound email.
6. Click **Save**.

What to do next:

- Allow users to reply to email notifications with text or attachments and post the reply as a response to the original message. For more information, see Enabling users to respond to Pulse email Notifications.
- Configure additional email accounts for each application for which you want to send Pulse notifications. For more information, see Configuring Pulse Notifications on additional email accounts.

©2025 Pegasystems Inc

Configuring Pulse notifications on additional email accounts

Enhance and categorize communication in your organization by creating multiple email accounts to send email notifications about Pulse messages. Consequently, users can quickly understand the context of the message, and as a result, process their work faster.

For example, you can configure an email account, such as HRServices@example.com, to send Pulse emails from the HR Services application, or SalesAutomation@example.com, to send Pulse emails from the Sales Automation application, so that users can quickly identify which application sends the notification.

> **Before you begin:**
>
> Create an email account for your application. For more information, see Configuring inbound email.

1. Copy the *pyPulseNotificationsEmailAccount* Data Transform that belongs to the *Pega-Social* Ruleset, to your application Ruleset:
   a. In the navigation pane of Dev Studio, click **Records**.
   b. Expand the **Data Model** category, and then click **Data Transform**.
   c. Click **Create**.
   d. In the **Label** field, enter a new name for the Rule.
   e. In the **Add to Ruleset** list, select your application Ruleset.
   f. Click **Create and open**.
2. On the **Definition** tab, in the **Source** column for the `Param.PulseNotificationsEmailAccount` target, enter your email account name, as in the following figure:

©2025 Pegasystems Inc

Data Transform for additional email accounts

3. Click **Save**.

Enabling users to respond to Pulse email notifications

Enhance communication and accelerate work processing by enabling an option to reply to Pulse messages by email. When users can post Pulse messages without logging in to an application, information exchange is faster and more convenient, and as a result, Cases reach resolution faster.

For example, in a sales automation scenario, a Pulse discussion occurs about converting a prospect to a customer in a Case. A sales person can quickly add comments to a post and attach a related document by replying to an email notification, without logging in to the application.

Before you begin:

- Create the *PulseNotifications* email account. For more information, see Configuring an email account for Pulse notifications.
- By default, the system looks for an email account with the name *PulseNotifications* so that it can process the Pulse notification emails. All applications in the system use this account to process their Pulse notification emails. If you want to configure a different email account for your application, see Configuring Pulse notifications on additional email accounts.

©2025 Pegasystems Inc

Users can receive Pulse notification emails in Cases that they follow and for which they choose to receive notifications. Users also receive emails when other users reference them in Pulse posts, or like their posts. You can allow users to quickly post attachments or text as replies by responding to emails instead of logging in to the application.

When users reply to emails, Pulse immediately fetches replies from Outlook, Gmail, Yahoo, iPhone, and Android email clients.

> **NOTE:** Users cannot post replies to Pulse by responding to the email notifications that they receive when other users like their messages.

To enable users to respond to Pulse email notifications, you need to create a relevant service package in your application, copy an existing Service email Rule and adjust it to your needs, and then create an email listener.

1. Create a service package in the application Ruleset:
   a. In the header of Dev Studio, click **Create** > **Integration-Resources** > **Service Package**.
   b. In the **Short description** field, briefly describe the purpose of the service package.

      > **For example:**
      >
      > Enter `Pulse email replies`.

   c. In the **Service Package Name** field, enter a label for the service package.

      > **For example:**
      >
      > Enter `PulseEmailReplies`.

   d. Click **Create and open**.

©2025 Pegasystems Inc

e. In the **Context** section, in the **Processing mode** list, select an option that matches your application, depending on whether your application is stateless or stateful.
f. In the **Service Access Group** field, enter the Access Group of your application.
   Ensure that the Access Group that you want to use can access service methods.
g. Clear the **Requires authentication** checkbox to bypass authentication and allow Pega Platform to invoke service methods.
h. To use TLS/SSL for service REST Rules that belong to this service package, select **Require TLS/SSL for REST services in this package**.
   For more information, see Defining processing and authentication for service packages.
i. Select the **Suppress Show-HTML** checkbox.
j. Click **Save**.

2. Create a service email Rule by copying an existing Rule:
   a. In the header of Dev Studio, click **Configure** > **Integration** > **Services** > **Service Definitions**.
   b. Expand **Service Email in package PulseEmailService class PegaSocial-Message**.
   c. Click **pzCreatePulseReply**.
   d. Click **Save as**.
   e. In the **Label** field, enter a new name for the Rule.
   f. In the **Customer Package Name** field, enter the name of the service package that you created in step 1.

      > **For example:**
      >
      > Enter `PulseEmailReplies`.

   g. In the **Customer class Name** field, enter a service class name.
   h. In the **Add to Ruleset** list, select your application Ruleset.

©2025 Pegasystems Inc

i. Click **Create and open**.
j. In the **Primary Page** section of the **Service** tab, ensure that the **Primary page class** field stores the `PegaSocial-Message` value.
k. In the **Service activity** section, ensure that the **Activity name** field stores the `pzPostReplyFromMail` value.
l. On the **Request** tab, ensure that the **Message header** and **Message data** sections store the same values as the *pzCreatePulseReply* Service email Rule.
m. On the **Response** tab, ensure that the value of the **Message type** list is `None` to avoid receiving blank emails when users reply to Pulse emails.
n. Click **Save**.

3. Create an email listener Rule:
   a. In the header of Dev Studio, click **Create** > **Integration-Resources** > **Email Listener**.
   b. In the **Short description** field, describe briefly the purpose of this listener.
   c. In the **Listener Name** field, enter a label for the listener.
   d. Click **Create and open**.
   e. On the **Properties** tab, from the **Startup option** list, select the appropriate option for running the listener, depending on your requirements.
   For more information about creating email listeners, see Creating an email listener.
   f. In the **Email Account** field, enter the email account for the listener that you use for sending emails with Pulse notifications.
   g. In the **Service information** section, in the **Service package** field, enter the service package that you created in step 1.
   h. In the **Service class** field, enter the name of the service class name that you entered in the **Customer class Name** field in step 2.
   i. In the **Service method** field, enter the name of the service email that you created in step 2.
   j. On the **Processes** tab, ensure that the **No Attachments** checkbox is cleared, so that attachments can be sent to email replies.
   k. Click **Save**.

PEGA ©2025 Pegasystems Inc

What to do next:

Manage your listeners in Admin Studio. For more information, see Managing listeners.

Configuring Pulse feed sources for Constellation

Ensure that users of your Constellation applications can view relevant messages by adding a feed source to their activity feed. For example, you can add an external feed source, so that customer service representatives (CSRs) that work with finances can see posts about currency rates.

Before you begin:

Create a feed source. For more information, see Creating feed sources for activity feeds.

1. In the navigation pane of Dev Studio, click **App**.
2. Click **Classes**.
3. In the text field, enter `Rule-Pulse`, and then open the *Rule-Pulse* Rule.
4. On the list of Pulse Rule instances, open the instance for which you want to edit the feed sources:
   - To edit how Pulse displays in your application dashboard, open *pyDashboardFeed*.
   - To edit how Pulse displays in Cases, open *pyCaseFeed*.
5. **Optional:** To add more feed sources, in the **Configure additional Pulse feeds** section, click **Add Pulse feed**, and then specify the feed source:
   a. In the **Name** field, enter the name of the feed source that you want to add.
   b. **Optional:** To display messages from the feed source every time a user opens a view that includes the Pulse gadget, select the **Display Pulse feed by default** checkbox.
   c. Click **OK**.

©2025 Pegasystems Inc

6. Click **Save**.

What to do next:

- Configure how the feed sources display in your Constellation application. For more information, see Configuring the display of feed sources.
- To make the feed source available for reuse in App Studio, mark it as relevant. For more information, see Marking a record as relevant.

- Configuring Pulse for Case Types
- Configuring the display of feed sources
- Posting a message in Pulse
- Collaborating with users by using Pulse
- Widgets in Constellation

Markdown in Pulse

You can use standard markdown syntax to format the text of your message in Pulse.

NOTE: After you submit your post, Pulse formats the text automatically using markdown syntax. If you use unsupported markdown syntax, the editor does not report any errors and renders the text as is.

The Pulse widget supports the following markdown syntax:

Use case	Syntax	Runtime
Headings H1 to H6	# Heading 1 ## Heading 2 ### Heading 3 #### Heading 4	

©2025 Pegasystems Inc

Use case	Syntax	Runtime
	##### Heading 5 ###### Heading 6	
Bold text	**bold**	**bold**
Italic text	*italics*	*italics*
Bold italics	***bold italics***	***bold italics***
Numbered list	1. First item 2. Second item 3. Third item	1. First item 2. Second item 3. Third item
Bullet list	- First item - Second item - Third item	• First item • Second item • Third item
Image	![Accessibility settings] (accessibility-icon.jpg)	
Case or user mention	@AnnaMiller	Anna Miller
Tag	#documentation	#documentation

The Pulse widget does not support the following markdown syntax:

- Block quote.
- Code.

©2025 Pegasystems Inc

- Horizontal rule.
- Multiple line breaks: the editor collapses multiple line breaks into a single break.

Related tasks

- Posting a message in Pulse
- Replying to a message in Pulse

Integrating Cases with external resources

Expand your Microjourney to take advantage of functionalities outside of your application, by connecting Cases to external resources. By expanding your processing to external resources, you can significantly speed up Case resolution.

For relevant training materials, see the Integrating with external applications module on Pega Academy.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

For example, you can increase the functionality of your application by assigning tasks to an external application or to a robotic queue.

- **Integrating a Case with an external resource**
- **Enabling Cases for creation through web and chat Channels**
- **Directed Web Access in configuring Assignments for external users**
- **Sourcing attachments from external storage**

©2025 Pegasystems Inc

- **Storing Case attachments using external storage**

Integrating a Case with an external resource

You can call an external resource, such as a robotic automation or another application, from a Process. 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 Process:

- **Assigning a task to a robotic Work Queue**
- **Assigning a task to an external application**
- **Connecting a Process to an external application**
- **Running a Decision strategy from a Process**
- **Calling an activity or Automation from a Process**

- Creating a stand-alone Process
- Calling one Process from another Process

Assigning a task to a robotic Work Queue

Integrate Pega Robotic Process Automation™ (RPA) in your Cases to define tasks that an unattended robot can carry out. As a result, you can reduce the need for human input in your application and resolve Cases faster.

For example, in a Case Type for requesting a new credit card, a customer service representative (CSR) puts the customer's call on hold whilst logging in to the system for ordering the printing of a new credit card and then switches to a second system for ordering the shipment of the newly created card. These operations can take up considerable time, distract the CSR from talking to the customer, and are prone to human error. Alternatively, the CSR can assign these tasks to a robotic Work Queue, which provides a quicker, more accurate service to the customer.

PEGA ©2025 Pegasystems Inc

For more information on assigning a task to a robotic Work Queue in Robot Manager, see Remote Integration

You assign a task to a robotic Work Queue by using the **Assign to robot queue** Step in a Process.

For more information on assigning a task to a robotic Work Queue in Robot Manager, see Remote Integration

For relevant training materials about using RPA, see the Pega Robotic Process Automation module on Pega Academy.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. In the Process in which you want to add a robotic Assignment, click **Add Step to the Process** > **More** > **Automations** > **Assign to robot queue**, and then click **Select**.

For example:

The following figure shows the Assign to robot queue Step in a Case Lifecycle:

©2025 Pegasystems Inc

Assign to robot queue Step in a Stage

3. Click the **Robotic Smartshape** icon.
4. **Optional:** To enable remote access, in the **Step** properties pane, on the **General** tab, select the **Enable remote access** checkbox.

> **Result:**
>
> The system displays the **Configure parameters** button, as described in step 7.

5. In the **Route to queue** field, enter the name of the Work Queue from which the robot retrieves Assignments.

> **Result:**
>
> The **Assign to robot queue** Step creates the Assignment that the robot retrieves.

©2025 Pegasystems Inc

6. In the **Run robotic Automation** field, enter the Automation that the robot runs after retrieving the Assignment.
7. **Optional:** To configure input and output parameters, edit the relevant properties:
   a. Click the **Configure parameters** button.
   b. Click the **Inputs** tab.
   c. In the **left pane**, select the required properties by clicking on one or more `+` symbols.

      > **Result:**
      >
      > For each selected property:
      > - In the left pane, the system displays a white tick in a green circle.
      > - In the central table, the system lists the selected properties with related information.

   d. Select the **Required** checkbox for each property that you require.
   e. Click the **Outputs** tab, and then complete steps 7.c and 7.d.
   f. Click **Submit** and then click **Save**.
8. **Optional:** To update the status of a Case by synchronizing it with the state of the robotic Assignment, click **Set Case Status**, and then, in the **Set Case Status** dialog box, for each of the Assignment states that you want to map to a Case Status, perform one of the following actions:
   - To select a status that a Case enters when a robotic Assignment reaches a certain state, press the Down arrow key in the corresponding field.
   - To create a custom status, enter a unique name in the field, and then click the **Open rule** icon.

> **NOTE:** If you do not select a status in the corresponding field, the Case keeps its previous state.

©2025 Pegasystems Inc

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 Assignment types in Pega Robot Manager. For more information, see Pega Robot Manager as a control center for your robotic workforce .
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 the validations that are specified for the Case.
Robotic Automation fails	Change Case status when a robotic Assignment cannot be completed and fails.
Robotic Automation has timed out	Change Case status when a robotic Assignment cannot be completed in the threshold time that is specified in the Assignment type Work Queue.

©2025 Pegasystems Inc

For example:

The following figure shows an example set of Case Statuses:

Scenario	Status
Assignment is queued	Pending-Robot
Robotic Automation is in progress	Pending-Automation
Robotic Automation has conflicting results	Pending-Conflict
Robotic Automation fails	Pending-Error
Robotic Automation has timed out	Pending-Timeout

Case Statuses mapped to robotic Assignment statuses

9. **Optional:** To validate the incoming values after your Automation is complete, in the Assignment Validation criteria field, select a Validation mode:

Choices	Actions
Use an existing Validation Rule	a. Select Existing Validation. b. In the list that appears below the Assignment Validation criteria field, select a Validation Rule. **NOTE:** You can select only a Validation Rule that is a relevant record. For more information, see Relevant records for Rule reuse.

©2025 Pegasystems Inc

Choices	Actions
Use custom Validation	a. Select **Custom Validation**. b. Click the **Click to configure Conditions** icon. c. In the **Configure Condition** window, specify how your application validates the Automation output.

If the Validation fails, the system routes the Assignment to the **Conflicting Robotic Assignments** Work Queue. If an Assignment fails, the system routes the Assignment to the Failed Robotic Assignments Work Queue.

10. **Optional:** To apply a goal and deadline to the robotic Assignment Step, on the **Goal & deadline** tab, in the **Assignment service level** field, press the Down arrow key, and then select the name of a Service-Level Agreement.
    For more information, see Setting Service-Level Agreements for Stages, Processes, and Steps.
11. Close the **Robot queue** dialog box by clicking **Submit**.
12. Click **Save**.
13. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

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

PEGA ©2025 Pegasystems Inc

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 Case Type manually.
- 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 Connect REST rules and Creating a REST integration by using the Create REST integration wizard.

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 Process instances, double-click a Process that you want to open.

©2025 Pegasystems Inc

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 Process 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 Process.
   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 Process 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.

©2025 Pegasystems Inc

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 Process 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 workflow.

3. Click **Submit**.
4. Click **Save**.

- Integrating a Case with an external resource
- Activity form - how to create activities for Processes

Adjusting a Service-Level Agreement for the Assignment Service shape

Meet your business goals by adjusting the Service-Level Agreement defined for the Assignment Service shape in a Process. By defining goals and deadlines, you can set expectations for the time that it takes an external system to complete a single task in the Process.

©2025 Pegasystems Inc

> **NOTE:** Service-Level Agreements that you define on shapes override Service-Level Agreements that are set on Processes and Case Types.

Service-Level Agreements on shapes override Service-Level Agreements that are set on Processes and Case Types.

1. Open the Process that contains the Assignment Service shape that you want to modify:
   a. In the navigation pane of Dev Studio, click **App**.
   b. On the **Classes** tab, expand the Case Type for which you want to add the Assignment, and then click **Process** > **Flow**.
   c. In the list of Process instances, double-click a Process that you want to open.
2. Modify the SLA for the Assignment Service shape:
   a. Double-click the Assignment Service shape to open the property pane.
   b. In the **Service level** field, press the Down Arrow key, and then select the name of a Service-Level Agreement.

   > **NOTE:** If you do not select an option, the system applies the *Work-.Default* Service-Level Agreement.

3. Click **Submit**.
4. Click **Save**.

- Adjusting initial Urgency for an Assignment
- Setting Service-Level Agreements for Stages, Processes, and Steps
- Creating a Service-Level Agreement Rule in Dev Studio

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

©2025 Pegasystems Inc

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 Process 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 Case Type manually.
- 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 Connect REST rules and Creating a REST integration by using the Create REST integration wizard.

The Integrator shape identifies an activity that connects your instance of Pega Platform™ to an external system to send or receive data. The Integrator shape runs synchronously, meaning that it must receive a response immediately, and then moves a Case to the next Step.

TIP: To improve efficiency, as a best practice use Data Pages to retrieve information from an external system. You can use a Data Page with an autopopulated property or with an autopopulated property backed by a Declare Expression. For more information, see Data Page definition.

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 Process instances, double-click a Process that you want to open.

©2025 Pegasystems Inc

d. On the form, on the **Diagram** tab, click the **Flow shapes** icon, and then click **Advanced Shapes** > **Integrator**.
e. Drag the Integrator shape to the Process 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** checkbox.
   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.

©2025 Pegasystems Inc

h. **Optional:** To activate a ticket in a Process 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 workflow.

3. Click **Submit** to close the property pane.
4. Click **Save**.

- Creating a stand-alone Process
- Creating an activity

Running a Decision strategy from a Process

You can use the **Run Data Flow** Decision shape to run a single Case Data Flow that runs a Decision strategy as part of the business process. By integrating Decision strategies with your Process, you create decision logic that is contextual for each Case.

Decision strategies combine business Rules and AI models to determine the most relevant and valuable action for each customer, with the goal of optimizing customer engagement and overall business results. In Pega Customer Decision Hub™, you can define the criteria for delivering Next Best Actions, such as channel limits, suppression Rules, engagement policies, and arbitration Rules for prioritizing Actions, by following the guided workflow in Next-Best-Action Designer. As you define the criteria, Next-Best-Action Designer generates the underlying Decision strategies for you. You can customize Decision strategies by using the designated extension points or by creating new Decision strategies manually. See also Strategy Result class.

©2025 Pegasystems Inc

1. Add the **Run Data Flow** shape to your Process.
   a. Open a Process by searching for it or by using the Application Explorer. For more information, see Finding Rules by class.
   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 Process diagram, based on the order of events in the Process.
   d. Connect the Run Data Flow shape, by dragging connector end points to connection points on different shapes in the Process.
2. Double-click the Run Data Flow shape to open the property pane.
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 Process.
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 pane.
7. Click **Save**.

- Creating a stand-alone Process
- Types of Data Flows
- Configuring a Strategy
- Process shapes

©2025 Pegasystems Inc

Calling an activity or Automation from a Process

Perform processing that does not require human interaction by calling an activity or Automation from a Process. As a result, you can create simpler, more efficient Processes that have fewer shapes.

For example, an insurance company is required to upload submitted insurance claims to a government registry of motor vehicles. As part of the Claim Registration Process, the application developer uses the Utility shape to call an activity or Automation that uploads the submitted claim.

> **Before you begin:**
>
> Create a utility activity or Automation that you want to call in your Case Lifecycle. For more information, see Creating an Automation and Creating an activity.

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 **Lifecycle** section, hover over a Stage, and then click **Configure Process**.
3. On the **Workflow** tab, click **Open Process in Dev Studio**.
4. On the **Diagram** tab, click the **Flow shapes** icon, and then click **Utility**.
5. Drag the Utility shape to a position on the Process diagram, based on the order of events in the Process.
6. Connect the Utility shape by dragging connector end points to connection points on different shapes in the Process.

Utility shape in a Process

7. Double-click the Utility shape to open the property pane.
8. In the **Select type of Rule** field, select the type of Rule.

©2025 Pegasystems Inc

9. Complete the following fields based on the type of Rule that you are using:

Choices	Actions
Run an Activity Rule **For more information, see Creating an activity.**	a. In the **Automation details** section, click Activity. b. In the Rule field, press the Down arrow key, and then select an activity of the Utility type. c. If the activity supports input parameters, in the additional fields, enter the values of the parameters that you want to set. **For example:** In an activity that runs a decision Data Flow, in the DataFlow field, enter the name of the Data Flow, such as `pzOfferFlow`, as shown in the following figure: *Example Data Flow parameter for an activity that runs a decision Data Flow*
Run an Automation Rule	a. In the **Automation details** section, click Automation.

©2025 Pegasystems Inc

Choices	Actions
For more information, see Automations.	b. In the **Resource** field, select the type of Automation that you want to use. c. In the **Name** field, select the Automation to use. d. If the Automation supports input parameters, in the additional fields, enter the values of the parameters that you want to set. **For example:** In an Automation that moves the Case to a specific Stage, in the **CaseID** field, enter the name of a Case, such as `JobApplication`, and in the **StageID** field, enter the name of a Stage, as shown in the following figure: *Example Case and Stage parameters for an Automation that moves a Case to a different Stage*

10. Click **Submit**.

©2025 Pegasystems Inc

11. Click **Save**.

- Process shapes
- Integrating a Case with an external resource
- https://academy.pega.com/topic/activities/v3
- https://academy.pega.com/topic/creating-activity/v2

Enabling Cases for creation through web and chat Channels

Ensure that users of your application can create Cases from outside Pega Platform™ by enabling Cases for creation through web and chat Channels. As a result, you help your customers reach their business goals in various scenarios that include different Channels and flexibility of processing.

For example, when you enable a Credit report Case Type for creation through a web Channel, a customer can start a new Case to create a Credit report directly through a website or a chat instead of logging in to a Pega Platform application.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. Click the **Settings** tab, and then open the **General** section.
3. In the **Availability** section, select the **Make available for adding to Create menu** checkbox.
4. Click **Save**.

For example:

The following image shows an example of enabling Cases for creation through web and chat Channels:

©2025 Pegasystems Inc

Enable Case creation through web and chat Channels

Related concepts

- Creating Cases with a web embed

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.

©2025 Pegasystems Inc

> ⚠ **IMPORTANT:** DWA is not supported in Constellation applications. For use cases requiring secure input from external users, consider using Pega's email channel with natural language processing (NLP) to ingest and process responses. This approach ensures authenticated, scalable handling of external communication while maintaining case context and data integrity. For more information, see Creating an Email Channel.

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:

PEGA ©2025 Pegasystems Inc

1. A Case reaches an external Assignment in the Lifecycle.
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 Lifecycle.
7. Your application updates the count of web invocations that are used in reports for license compliance.

- **Configuring Directed Web Access**

- Sending automatic emails from Cases
- Requesting approval by email and push notifications

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.

> ⚠ **IMPORTANT:** DWA is not supported in Constellation applications. For use cases requiring secure input from external users, consider using Pega's email channel with natural language processing (NLP) to ingest and process responses. This approach ensures authenticated, scalable handling of external

©2025 Pegasystems Inc

communication while maintaining case context and data integrity. For more information, see Creating an Email Channel.

Before you begin:

- To give an external party a one-time DWA to perform an action, ensure that the Process in which you want to use DWA already exists. For more information, see Creating a Case Type manually.
- Create the operator ID for external users. Ensure that the operator ID is `External` and the Access Group is `PRPC:Unauthenticated` as in the following figure:

Adding a new operator ID for the external user

For more information, see Creating an operator ID.

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.

©2025 Pegasystems Inc

For example:

The following figure shows adding an Access Group for the external user:

> **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** checkboxes 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.

©2025 Pegasystems Inc

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** > **Email Accounts**.
   b. Create an email account in the appropriate class group.
   For more information, see Configuring outbound email.
   c. Click the **Test connectivity** link and ensure that you have the outbound email Rule for the application running.
4. Configure a Correspondence Rule that contains the DWA URL:
   a. Create a Correspondence Rule of the *Rule-Obj-Corr* type.
   For more information about how to open a Create Rule form, see Creating Rules.
   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**.

©2025 Pegasystems Inc

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 **Cases**, 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:

©2025 Pegasystems Inc

For more information about how to configure a Case participant, see Defining Case participants.

d. In the **Data transform** field, enter a name for a new Data Transform.
e. In the Data Transform configuration form, on the **Definition** tab, in the **Target** field, enter `.pyWorkPartyUri`.
   Each party represented in a work object has a unique identifier, recorded in the *pyWorkPartyUri* standard property. Depending on the class or other processing, it can be an operator ID, email address, account number, or other information.
f. In the **Source** field, enter the operator ID for the external user.

For example:

The following figure shows a Data Transform for mapping a user identifier with the external user role:

g. Click **Save**.
h. Select the **Create participant automatically when the Case starts** checkbox.
i. Click **Done**.
j. Click **Save**.

6. Configure the Assignment shape:
   a. In the navigation pane of Dev Studio, click **App**.
   b. On the **Classes** tab, expand the Case Type in which you want to edit the Assignment, and then click **Process** > **Flow**.
   c. In the **Flow** form, on the **Diagram** tab, double-click an Assignment.

©2025 Pegasystems Inc

d. In the **Assignment properties** dialog box, in the Routing section, from the Route to list, select Custom.
e. In the Assignment type list, select External.
f. In the Router field, select ToWorkbasket, and then provide the name of the Work Queue to which you want to route the Assignment.
g. In the Advanced section, click the Assignment details node, and then, in the OperatorModel field, select the operator ID for the external user.
h. In the DaysToExpiration field, enter the number of days after which the DWA link expires and the external user can no longer open the Assignment.
i. In the Party field, enter the operator ID, for example `External`.
j. In the Subject field, enter the email message subject.
k. In the HarnessPurpose field, select PerformExternal.
l. Click Submit.
m. Save your changes by clicking Save.

7. Define what content is displayed to users after the Assignment is completed, by configuring the appropriate post-processing behavior of the Flow Action that is associated with the Assignment:
   a. In the navigation pane of Dev Studio, click Records.
   b. Expand the Process category, and then click Flow Action.
   c. Open the Flow Action Rule.
   d. On the Action tab, in the **Post-processing** section, navigate to the **Back-to-back processing configuration** subsection.
   e. In the If an Assignment is not being performed list, select Show Harness, and then in the text field, select the *ConfirmExternal* harness.
   f. Click Save.

   For more information, see Configuring back-to-back processing for a Flow Action.

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.

©2025 Pegasystems Inc

> The system creates the external Assignment. The flow pauses until the external party completes the external Assignment. The link in the external Assignment does not require login, but the user can use it only once.

Related concepts

- Operators
- Flow Actions

Related tasks

- Creating an operator ID

Related reference

- Correspondence rules and email
- Process shapes

Sourcing attachments from external storage

Gain access to third-party documents and enhance collaboration between stakeholders by configuring external storage as a source of content that you can attach to your Case, space, or Pulse conversation.

Sourcing from external storage enables you to use more relevant data in your business process. For example, you can configure an external source for attachments for customer service representatives who manage loan requests to ensure that they use the most up-to-date version of the necessary documents, such as the Terms and Conditions document.

©2025 Pegasystems Inc

Before you begin:

- To enable the sourcing of content from web storage providers, ensure that web storage providers are correctly configured. For more information, see Installing components.
- To enable the sourcing of content from file storage repositories, create or configure a repository in your system. For more information, see Creating a repository.
- To enable the sourcing of content from Content Management Interoperability Services (CMIS) systems:
  - Ensure that you have a connector configured to support interaction between your system and a CMIS system. For more information, see Integration Connectors landing page and Integration-Connectors category.
  - Ensure that you have a *Connect-CMIS* Rule configured in your system. For more information, see Creating a Connect CMIS Rule.

> ⚠ **IMPORTANT:** Ensure that you review the restrictions on adding a repository for Case attachments in Requirements and restrictions for Case attachments in a file storage repository. If you fail to observe restrictions while you implement a repository, you might experience system failures.

As a best practice, use repositories for Case attachments. Repositories use data streaming to present attachment data directly to your application. When you use web storage or CMIS, Pega Platform™ places data on the Pega Clipboard before use in the application, which uses more system resources.

Your environment type determines the repositories that you can configure for your external content for Case, space, or Pulse attachments:

PEGA ©2025 Pegasystems Inc

- In Pega Cloud environments, you can use `pegacloudfilestorage` and the `/attachments` subfolder as the repository for Case attachments, or you can configure another type of repository.

  Before integrating the repository in your application, you can change the repository that you use for Case attachments to another external repository type after you create the repository.

  For more information, see Creating a repository.

- In on-premise and client-managed cloud systems, you specify an external repository for Case attachments after you create the repository.

  For more information, see Creating a repository.

> ⚠ **IMPORTANT:** Do not use the internal Pega Platform database as a repository.

1. In the header of Dev Studio, click the name of the application, and then click **Definition**.
2. On the Application form, click the **Integration** tab.
3. In the **Content sourcing** section, configure at least one external storage system:

Choices	Actions
A web service provider	a. Select the **Source from web storage providers** checkbox. b. In the **Provider** list, select an installed web storage provider for which you have an account and a content location. c. In the **Authentication profile** box, enter or select an Authentication Profile of the OAuth 2.0 type.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: Use the same Authentication Profile to store content in and source from the web storage provider. d. **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. e. **Optional:** To use content from more than one web storage provider, click **Add**, and then repeat these steps to configure another provider.
A repository	a. Select the **Source from repositories** checkbox. b. In the **Repository** list, select a repository for which you have an account and a content location. **NOTE:** Pega Cloud clients: To use Pega Cloud File storage, select the `pegacloudfilestorage` repository.

©2025 Pegasystems Inc

Choices	Actions
	c. Click **Browse**, and then select the folder from which to source attachments. **NOTE:** Pega Cloud® clients: To use Pega Cloud File storage, select the `/attachments` sub-folder. d. Click **Select**. e. **Optional:** To change the label that users see when they add an attachment from this repository, in the **Label** field, enter a description. f. **Optional:** To use content from more than one folder, click **Add**.
CMIS systems	a. Select the **Source from CMIS systems** checkbox. b. **Optional:** To change the label that users see when they add an attachment from this source, in the **Label for this content source (where available)** field enter a description.

For example:

©2025 Pegasystems Inc

Application configured to enable the sourcing of content for attachments from a web storage provider and a repository

4. Click Save.

Result:

Your content from external storage becomes available when you add an attachment to a Case, space, or Pulse conversation.

- File and content storage
- Storing Case attachments using external storage
- Attaching content to a Case
- Attachment types
- Creating an authentication profile
- Integrating your application with external resources

Adding attachments from external locations

Provide more data and relevant context to your work by supplementing your Case, space, or Pulse conversation with external content.

©2025 Pegasystems Inc

Log in to a Portal, and then add an attachment from an external location:

Choices	Actions
Add a Case attachment	a. Create or find a Case to which you want to attach external content. **NOTE:** You can locate a Case by searching for it or by looking in your Worklist or the **Recents** list. For more information, see Creating Cases from the navigation pane of a Portal. b. In the upper-right corner, click the **Expand the utility pane** icon. c. In the **Files & documents** section, click the **Manage files and documents** icon, and then in the **Manage content** dialog box, click the menu option from which you can add files from an external system. The options in the menu show the values that you provide in the **Label** field on the **Integration** tab of your application rule. For more information, see Sourcing attachments from external storage.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: When you connect to your external storage account from the application for the first time, you might receive a prompt to authenticate and authorize the connection. **For example:** *The external storage options configured for the application shown in the menu* d. Attach a file from your external system. **NOTE:** Attaching multiple files with the same name to the same Case results in an error because storage providers treat the file name together with the path as a key. e. Click **Submit**.

©2025 Pegasystems Inc

Choices	Actions
	Result: The Case attachment is displayed in the **Files & documents** section. **NOTE:** To remove a Case attachment, click the Delete icon next to the name of the attachment.
Add a space attachment	a. In the navigation pane, click Spaces. b. Click the space to which you want to add content. c. In the upper-right corner, click the Expand the utility pane icon, and then click the **Related** tab. d. In the **Files & documents** section, click the Manage files and documents icon, and then in the **Manage content** dialog box, click the menu option from which you can add files from an external system. The options in the menu show the values that you provide in the Label field on the **Integration** tab of your

©2025 Pegasystems Inc

Choices	Actions
	application rule. For more information, see Sourcing attachments from external storage. **For example:** *A menu with the external storage options configured for the application* e. Attach a file from your external system. **NOTE:** Attaching multiple files with the same name to the same Case results in an error because storage providers treat the file name together with the path as a key. f. Click **Submit**. **Result:** The space attachment is displayed in the **Files & documents** section.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: To remove a space attachment, click the **Delete** icon next to the name of the attachment.
Add a Pulse attachment	a. Find or create a Pulse message to which you want to add content. For more information, see Viewing your activity feed and Posting a message in Pulse. b. In the field that is displayed in the **Pulse** section, click **Add attachments** > **External Sources**, and then click an option from which you can add files from an external system. The options in the menu show the values that you provide in the **Label** field on the **Integration** tab of your application rule. For more information, see Sourcing attachments from external storage.

©2025 Pegasystems Inc

Choices	Actions
	For example: *Attaching files from external locations in Pulse* c. Attach a file from your external system. d. Click **Post**. **Result:** The Pulse attachment is displayed in the **Pulse** section.

Adding Case attachments from a CMIS system

Enable users to access content stored in external sources by adding an attachment from a CMIS system to a Case. Downloading content from an external CMIS system guarantees that users use the most up-to-date version of documents.

Before you begin:

- Enable sourcing from CMIS systems on the application Rule form. For more information, see Sourcing attachments from external storage.

©2025 Pegasystems Inc

- Ensure that you have a connector configured to support interaction between your system and a CMIS system. For more information, see Integration Connectors landing page and Integration-Connectors category.
- Ensure that you have a *Connect-CMIS* Rule configured in your system. For more information, see Creating a Connect CMIS Rule.

Content Management Interoperability Services (CMIS) are external systems that use the CMIS standard, such as Alfresco, Microsoft SharePoint, and OpenText, to which you connect by using a CMIS Connect Rule. CMIS systems support storing and sourcing content. For more information, see About Connect CMIS rules.

You can use any CMIS-compatible server to source data for your application.

1. Log in to a Portal, and then create or find a Case or space to which you want to attach a file from a CMIS system.
2. In the upper-right corner, click the **Expand the utility pane** icon.
3. In the **Files & documents** section, click the **Manage files and documents** icon, and then in the **Manage content** dialog box, click the **Content from CMIS** tab.
4. In the **Select a file** section, select a file from one or more CMIS systems.
   - The CMIS systems in the list correspond to the *Connect-CMIS* Rules that are available for the current operator.
   - The files in a CMIS system come from the CMIS repository that you configure in your *Connect-CMIS* Rule that references the system.
5. In the **Name** field, provide a name for the attachment.
6. Click **Submit**.

   Result:

   The attachment is displayed in the **Files & documents** section.

7. **Optional:** To remove a Case attachment, click the **Show more** icon next to the name of the attachment, and then click **Delete**.

©2025 Pegasystems Inc

Managing Case attachments stored in a CMIS system

Manage versions of a Case attachment that is stored in a CMIS system at run time to control which version of the attachment you want users to download.

1. Click the **Gear** icon next to the name of the attachment.
2. In the **Manage attachment** window, in the **Version history** section, click the file name for the appropriate version of the attachment to download.
3. **Optional:** To prevent other users from modifying the file attachment, select **Lock this attachment for local editing**.
4. Click **Upload new version** to check in the new version of the file.

Result:

The new version is displayed in the **Version history** section.

Storing Case attachments using external storage

Upload large files without causing out-of-memory errors by configuring your application to store Case and Pulse attachments outside the Pega Platform™ database. For example, you can configure an external storage for attachments for insurance agents to ensure that the system always have enough memory even with heavy upload of files that support insurance claims.

Before you begin:

- To enable storing of content by using web storage providers, ensure that you correctly configure your web storage providers. For more information, see Installing components.
- To enable storing of content by using file storage repositories, create or configure a repository in your system. For more information, see Creating a repository.

©2025 Pegasystems Inc

- To enable storing of content by using Content Management Interoperability Services (CMIS) systems, perform the following activities:
  - Ensure that you configure your connector to support interaction between your system and a CMIS system. For more information, see Integration Connectors landing page and Integration-Connectors category.
  - Ensure that you configure a *Connect-CMIS* Rule in your system. For more information, see Creating a Connect CMIS Rule.

> ⚠ **IMPORTANT:** Ensure that you review the restrictions for adding a repository for Case attachments in Requirements and restrictions for Case attachments in a file storage repository. If you fail to observe restrictions as you implement a repository, you can experience system failures.

The default storage for on-premises deployments is the Pega database. As a best practice, use repositories for Case attachments. Repositories are the only supported content storage type that supports streaming and that has no risk of running out of memory.
Your environment type determines the repositories that you can configure for your external content for Case or Pulse attachments:

- In Pega Cloud environments, you can use `pegacloudfilestorage` and the `/attachments` subfolder as the repository for Case attachments, or you can configure another type of repository.

  Before integrating the repository in your application, you can change the repository that you use for Case attachments to another external repository type after you create the repository.

  For more information, see Creating a repository.

PEGA ©2025 Pegasystems Inc

- In on-premise and client-managed cloud systems, you specify an external repository for Case attachments after you create the repository.

  For more information, see Creating a repository.

> ⚠ **IMPORTANT:** Do not use the internal Pega Platform database as a repository.

For more information about different types of content storage types that you can use, see Integrating with file and content management systems and File and content storage.

> ⓘ **NOTE:** To access the contents of a repository that stores attachments, you can use the *D_pxGetFile* API.
>
> For more information, see Repository APIs and Using repository APIs in your application.

1. In the header of Dev Studio, click the name of the application, and then click **Definition**.
2. On the Application form, click the **Integration** tab.
3. In the **Content storage** section, configure at least one external storage system:

Choices	Actions
CMIS systems	a. Select the **Store in CMIS system** checkbox. b. In the **Connector name** field, select a connector.
A web service provider	a. Select the **Store in web storage provider** checkbox.

©2025 Pegasystems Inc

Choices	Actions
	b. In the **Provider** list, select an installed web storage provider for which you have an account and a content location. c. In the **Authentication profile** box, enter or select an Authentication Profile of the OAuth 2.0 type. **NOTE:** Use the same authentication profile to store content in and source from the web storage provider. d. Click **Browse**, and then select the target folder for your attachments. e. Click **Select**. f. **Optional:** To display a prompt to users before they delete an attachment, click the **Prompt users for confirmation before deleting content from storage location** checkbox.
A repository	a. Select the **Store in repository** checkbox. b. In the **Repository** list, select a repository for which you have an account and a content location.

©2025 Pegasystems Inc

Choices	Actions
	NOTE: Pega Cloud clients: To use Pega Cloud File storage, select the `pegacloudfilestorage` repository. c. Click **Browse**, and then select the folder for your attachments. **NOTE:** Pega Cloud® clients: To use Pega Cloud File storage, select the `/attachments` sub-folder. d. Click **Select**. e. **Optional:** To display a prompt to users before they delete an attachment, click the **Prompt users for confirmation before deleting content from repository** checkbox.

For example:

©2025 Pegasystems Inc

Application configured to enable the storing of content for attachments in a repository

4. Click **Save**.

Result:

Pega Platform adds new attachments that you create in a Case or Pulse conversation to the location that you choose.

- Integrating with file and content management systems
- File and content storage

Limitations for attachments in a file storage repository

When you configure your application to store Case or Pulse attachments in a repository, consider the limitations to ensure that your attachments are stored and retrieved correctly.

Ensure that you review the following limitations for adding a repository for Case attachments:

©2025 Pegasystems Inc

- Repositories support only file attachments. All other types of attachments reside in a database.
- You cannot change the content storage type from repository to a different type.
- After you switch from database storage to repository storage, you cannot switch back to database storage without losing access to the attachments in the repository.
- You cannot modify the configuration of a repository after the repository is in use.
- Application attachments that you provide on the application form reside in a database.
- Multiple applications in the same application Rule stack must have the same repository configuration, or users cannot access the file attachments.
- Because the application Pulse gadget cannot prompt for authentication details, Pulse attachments that you provide on the Application Overview landing page reside in the repository only when your session is already authenticated or when the repository supports non-interactive authentication. If both of these conditions are false, an error occurs and the operation fails.
- The following features are not supported for custom repositories during DevOps:
  - Import and export
  - File listener
- When there are multiple image or PDF attachments, poster creation is skipped, and Pega Platform™ generates a placeholder thumbnail instead.
- When using database storage, you can attach multiple files that have the same name to the same Case. For external storage, you cannot attach multiple files that have the same name to the same Case because storage providers treat the file name plus the path as a key. Attempting to attach multiple files that have the same name to a Case causes an error. You can, however, attach files that have the same name to different Cases.

For more information, see Creating a repository and File and content storage.

> **NOTE:** When you configure your application to store Case attachments in a repository, Pega Platform appends the Case ID to the attachment

©2025 Pegasystems Inc

> name. For example, an attachment called `loanapplication.pdf` becomes `loanapplication_CASE-1234.pdf`.

- The unsupported characters that might not work in attachment names are {"""", "?", "*", "<", ">", "|", ":"} .

Migrating Case attachments to a repository

Reduce the size of your Pega Platform™ database and improve performance by migrating the Case attachments for your application from a Pega Platform database storage to a repository.

> **NOTE:** When you migrate your applications from on-premises to Pega Cloud® deployments, move your attachments from a Pega database to a repository.

> **Before you begin:**
>
> Ensure that the *pzAttachmentMigrationProcessor* queue processor is running. For more information, see Changing a queue processor Rule state.
>
> > **NOTE:** Note that *pzAttachmentMigrationProcessor* is configured to run for a maximum of 100,000 attachments. For migrating a larger number of attachments, please create a custom queue processor that suits your needs.

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.

©2025 Pegasystems Inc

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:

After the migration finishes, the results include the number of migrated files. For the files that were not migrated, the system work object ID and a description of the error, which you can use for troubleshooting.

What to do next:

Ensure that the repository to which the files were migrated is the same repository where you store new attachments. For more information, see Sourcing attachments from external storage.

Designing questionnaires

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

NOTE: As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

For more information, see Creating a new application from a Blueprint.

By taking advantage of the multiple formats that questions can take in your questionnaires, you ensure that you capture the precise information that your business

©2025 Pegasystems Inc

processes require. For example, you can provide a text box in which users can enter their answers directly, or create a radio button matrix, so that users can select their answers from a group of radio buttons. To save time, you can collect related questions on one page, and then reuse that page in other questionnaires. For instance, you can create a question page that captures details about customer satisfaction, and then reuse that question page in questionnaires about different business processes.

The following figure shows the process of authoring a questionnaire that collects a customer's answers about their stay at a spa resort, and the same questionnaire at run time:

Designing a questionnaire and running it in your application

- **Creating a questionnaire**
- **Running a questionnaire in a Case**
- **Fine-tuning your questionnaire**
- **Visualizing questionnaire data by creating Insights in Constellation**
- **Using advanced questionnaire features in Dev Studio**

©2025 Pegasystems Inc

- Running a questionnaire in a Case
- Feature changes and limitations in Constellation

Creating a questionnaire

Collect a wide range of information from users by providing questionnaires in a low-code way in App Studio. By including questions in different formats, you ensure that you gather the precise data that you need for your business Processes. For example, after a customer service representative (CSR) finishes processing an insurance claim review Case for a customer, the customer can complete a questionnaire to provide feedback on the CSR's performance.

Questionnaires store answers to questions in data objects in a Data Model. You can reuse data objects across your application. For example, if a customer provides a Case ID in a questionnaire, you can use the data object that stores the Case ID in other, related Cases. To speed up the creation of your questionnaire, you can then upload an `.xlsx` or `.csv` file that contains a Data Model.

1. In the navigation pane of App Studio, click **Case Types**.
2. In the header of the **Case Types** landing page, click **New**.
3. In the **Create Case Type** dialog box, in the **Case Type name** field, enter the name of your questionnaire.

   > **For example:**
   >
   > Enter `Feedback on CSR`.

4. Expand the **Advanced** section, and then, in the **Type** section, select **Questionnaire**.
5. **Optional:** To speed up the development of your questionnaire, reuse assets from another Case Type:
   a. Expand the **Advanced** section.
   b. In the **Reuse assets from** list, select a Case type that you want to reuse.

©2025 Pegasystems Inc

6. **Optional:** To provide fields for your questionnaire upon creation, upload a file that contains a Data Model:
   a. Expand the **Advanced** section, and then select the **Create Data Model using spreadsheet** checkbox.
   b. Click **Next**.
   c. In the **Select file** dialog box, click **Choose File**.
   d. Navigate to the file with the Data Model, and then click **Open**.
   e. Click **Next**.
   f. In the **Create fields** dialog box, specify field names and field types, and then click **Create**.
7. Click **Save**.

What to do next:

Populate your questionnaire with questions. For more information, see Adding a question page to a questionnaire.

- **Adding a question page to a questionnaire**
- **Adding a question to a question page**
- **Best practices for creating questionnaires**

- Extending a Data Model for a Case
- Running a questionnaire in a Case
- Best practices for creating questionnaires

Adding a question page to a questionnaire

To organize related questions, create a question page. You can reuse a question page across multiple questionnaires, and as a result, speed up the development of your application.

PEGA ©2025 Pegasystems Inc

For example, you can create a question page to store questions about the service at a hotel. Then you can reuse the question page for questionnaires that gather answers about the quality of the hotel restaurant, and the room service. When customers complete the questionnaires, they interact with different questions about the restaurant and the hotel, but the questions about the service quality are the same.

Before you begin:

- Create a questionnaire. For more information, see Creating a questionnaire.
- Create a Data Model for your questionnaire. For more information, see Extending a Data Model for a Case.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. On the **Questionnaire** tab, add a question page:
   - To create a new question page, click **Add questionnaire item** > **Question page**.

©2025 Pegasystems Inc

- To reuse an existing question page, click **Add questionnaire item** > **Library question page**, and then, in the **Questionnaire library** dialog box select a question page to add, and click **Add**.

3. In the **Title** field, click **Click here to edit page title**, and then enter a unique name for the question page, for example `Service quality`.
4. **Optional:** To display the question page only in specific scenarios, define Visibility conditions.
   For more information, see Conditionally displaying a question page in a questionnaire.
5. **Optional:** To provide additional guidance for users, add instructions that an application displays at the top of the question page:
   a. In the **Page settings** pane, in the **Instructions** section, click the **Properties** icon.
   b. In the rich text editor, enter your text.

      > **For example:**
      >
      > Enter `Rate the quality of service.`

   c. Click **OK**.
6. **Optional:** To manage your question pages in an organized way, categorize the question page:
   a. In the **Page settings** pane, in the **Category** section, click the **Properties** icon.
   b. In the **Category** field, enter the category that you want to associate with the question page, and then click **OK**.
7. Save your changes by clicking **Save**.
8. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

What to do next:

Populate your question page with questions. For more information, see Adding a question to a question page.

- Conditionally displaying a question page in a questionnaire
- Running a questionnaire in a Case

Adding a question to a question page

Prompt users for information in a format that you define by adding a question to a question page. At run time, users interact with questions that you provide and deliver data that your business processes require to reach resolution. By engaging multiple types of questions, you can collect the exact information that you need. For example, you can create a text box question and ask users to provide their name.

Before you begin:

Create a question page to store the questions. For more information, see Adding a question page to a questionnaire.

For improved management of data in your application, you associate the questions with fields from a Data Model. The fields store the answers to the questions that users provide at run time. To meet your precise business needs, you can mark questions as required to ensure that users enter these answers before submitting the questionnaire. You can also validate the answers to avoid issues and help users provide information in an expected format.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. On the **Questionnaire** tab, on the question page that you want to edit, add a question:

Choices	Actions
Create a new question	a. Click **Add question** > **New**. b. In the list of question types, select a type that you want to use, for example, select **Text (paragraph)**. c. In the text editor, enter the text of the question. **For example:** Enter `What would you change in our customer service?` d. If a question type that you want to create requires additional formatting, provide relevant configurations. For more information about formatting questions, see:

©2025 Pegasystems Inc

Choices	Actions
	• Formatting text boxes in a questionnaire. • Formatting a picklist in a questionnaire. • Formatting a checklist in a questionnaire. • Formatting a slider in a questionnaire. • Formatting a radio button matrix in a questionnaire. • Formatting a currency question in a questionnaire. • Formatting a decimal question in a questionnaire. **NOTE:** Only traditional UI Pega Platform applications support most questions that require formatting. Applications based on Constellation do not display formatted questions. A picklist is the only formatted question that you can use both in standard Pega Platform and Constellation applications. You can use currency and decimal

©2025 Pegasystems Inc

Choices	Actions
	questions only in Constellation applications.
Reuse an existing question	a. Click Add question > From library. b. In the **Questionnaire library** dialog box, select a question that you want to reuse, and then click Add next to the question. c. Close the dialog box by clicking Add.

3. In the Map answer to field list, select a field in the Data Model that stores the answer:

Choices	Actions
Reuse an existing field	In the list, select the field from a Data Model.
Create a new field	a. In the list, click Create New. b. In the **Add field** dialog box, in the Field name field, enter the name, for example, enter `SatisfactionLevel`. The field type is the same as the question type. c. **Optional:** To provide additional configuration for the field, expand the Advanced section, and then provide a new field ID, a description, and the maximum number of characters of the value that the field stores.

©2025 Pegasystems Inc

Choices	Actions
	d. Click **Submit**.

4. **Optional:** To ensure that users answer a question, mark the question as required. For more information, see Making a question required in a questionnaire.
5. **Optional:** To ensure that users provide answers in an expected format, create a Validation.
   For more information, see Validating answers in a questionnaire.

   > **Result:**
   >
   > If the Validation fails at run time, users receive an error message. Users need to correct the answers with failed Validation before submitting the questionnaire.

6. **Optional:** To display the question only in selected scenarios, configure Visibility Conditions.
   For more information, see Conditionally displaying a question in a questionnaire.

   > **Result:**
   >
   > At run time, the questionnaire displays the question only if the Visibility Conditions evaluate to true.

7. **Optional:** To manage your questions efficiently, categorize the question:
   a. In the **Question settings** pane, in the **Category** section, click the **Properties** icon.
   b. In the **Category** field, enter the category, for example, `CustomerService`.
   c. Click **OK**.
8. **Optional:** To add more questions to the question page, repeat steps 2 through 7.
9. Click **Save**.
10. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

©2025 Pegasystems Inc

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

What to do next:

Configure how your questionnaire displays at run time. For more information, see Creating Views for Case Types.

- **Formatting text boxes in a questionnaire**
- **Formatting a picklist in a questionnaire**
- **Formatting a checklist in a questionnaire**
- **Formatting a slider in a questionnaire**
- **Formatting a radio button matrix in a questionnaire**
- **Formatting a currency question in a questionnaire**
- **Formatting a decimal question in a questionnaire**
- **Scoring answers in a questionnaire**

- Creating questions in bulk
- Creating a stand-alone question with a single answer
- Defining Conditions in the Condition builder
- Running a questionnaire in a Case

Formatting text boxes in a questionnaire

Format multiple text boxes in a questionnaire to ask users to provide short text answers. For example, in a questionnaire for job candidates, you ask users to provide

©2025 Pegasystems Inc

their university major, previous employer, and a name of their own business if they have one.

Before you begin:

Add a question of the multi-text boxes type to your questionnaire. For more information, see Adding a question to a question page.

NOTE: Only traditional UI Pega Platform applications support questions with multiple text boxes. Applications based on Constellation do not display questions with multiple text boxes.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Multi-text boxes**.
4. In the text field, enter the text of the question.
5. Label the text boxes:

©2025 Pegasystems Inc

   a. In the **Textbox options** section, click Add value.
   b. In the text field, enter a label for the text box.
6. **Optional:** To display the text box only in selected situations, define Conditions:
   a. Click More > Advanced view.
   b. From the Visible when list, select When, and then click Edit.
   c. Provide a value to compare at run time, a comparator, and a value to compare against the first value.

      > **For example:**
      >
      > To display a text box to provide a company name only to users who have their own businesses, create the condition `HasCompany equals to true`.

   d. **Optional:** To add more Conditions, click Add a row.
   e. If you create multiple rows, define whether an answer needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
   f. Click Submit.
7. Click Save.
8. **Optional:** To view how your changes are reflected at run time, click Preview and run the Case.

> **IMPORTANT:** Avoid using Save and run to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **For example:**
>
> The following figure shows a sample multi-text boxes question at run time:

©2025 Pegasystems Inc

Multi-text boxes question

- Designing questionnaires
- Defining Conditions in the Condition builder
- Fine-tuning your questionnaire

Formatting a picklist in a questionnaire

Format a picklist in a questionnaire to define the choices that users can select to answer a question. With a picklist, users can select only one answer to a question. For example, you can ask users about their rating of customer service in your company, and then provide a choice of answers that range from excellent to poor.

Before you begin:

Add a question of the picklist type to your questionnaire. For more information, see Adding a question to a question page.

1. Navigate to the questionnaire:

©2025 Pegasystems Inc

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the `Questionnaire` type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of , click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Picklist**.
4. In the **Question settings** pane, in the **Map answer to field** list, select a field of the picklist type:

Choices	Actions
Reuse an existing field	In the list of fields, select the field that you want to use.
Create a new field	a. In the list of fields, click **Create new**. b. In the **Add field** dialog box, in the **Field name** field, enter a name for the field in a Data Model. c. In the dialog box, in the **Display as** list, define a display mode for the picklist. You can display a picklist as a drop-down list or a list of radio buttons. d. In the **Picklist** options, define choices for the picklist.

©2025 Pegasystems Inc

Choices	Actions
	You can create your own local choices or source them from a Data Page. e. Click **Submit**.

5. Click **Save**.
6. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a picklist displayed as radio buttons that users use to rate customer service:

Picklist with radio buttons

©2025 Pegasystems Inc

What to do next:

Provide tools to compare and calculate weight of responses by scoring answers in the picklist. For more information, see Scoring answers in a questionnaire.

- Designing questionnaires
- Fine-tuning your questionnaire

Formatting a checklist in a questionnaire

Format a checklist in a survey to define the choices that users can select to answer a question. With a checklist, users can select multiple answers to a question. For example, in a questionnaire for job applicants, you can ask users to select the languages that they speak.

Before you begin:

Add a question of a checklist type to your survey. For more information, see Adding a question to a question page.

NOTE: Only traditional UI Pega Platform applications support questions with a checklist. Applications based on Constellation do not display questions with checklists.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.

©2025 Pegasystems Inc

Choices	Actions
	b. In the list of Case Types, open a Case Type of the `Questionnaire` type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Survey**. b. Click the questionnaire that you want to edit.

2. If you have multiple questions on the question page, click the question that you want to edit.
3. From the list of question types, select **Checklist**.
4. In the rich text editor, enter your question.

> **For example:**
>
> Enter `Select languages that you speak.`

5. Define the choices in the checklist:
   a. In the **Checklist options** section, click **Add value**.
   b. In the text field, enter a label for the checkbox.

   > **For example:**
   >
   > Enter `French`.

6. **Optional:** To control when the checklist displays a choice, define conditions:
   a. Click **More** > **Advanced view**.
   b. In the **Visible when** list, select **When**, and then click **Edit**.
   c. Provide a value to compare at run time, a comparator, and a value to compare against the first value.

©2025 Pegasystems Inc

> **For example:**
>
> To display `French` only to users who apply for a job in a sales department, create the Condition `Position is equal to Sales`.

d. **Optional:** To add more Conditions, click **Add a row**.
e. If you create multiple rows, define whether an answer needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
f. Click **Submit**.

7. **Optional:** To store the selected choice under a name that is different from the label, change the reporting value:
   a. Click **More** > **Advanced view**.
   b. In the **Reporting value** field, replace the default value with your custom value.
      For example, in a list of languages that users speak, you can use an abbreviation for each language.
8. Click **Save**.
9. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **For example:**
>
> The following figure shows a sample checklist at run time:

©2025 Pegasystems Inc

Checklist with multiple options

What to do next:

Provide tools to compare and calculate weight of responses by scoring answers in the checklist. For more information, see Scoring answers in a questionnaire.

- Using advanced questionnaire features in Dev Studio
- Defining Conditions in the Condition builder
- Fine-tuning your questionnaire

Formatting a slider in a questionnaire

Format a slider in a survey to control the range of values from which users select an answer. For example, you can ask how many days users work each week by formatting a slider with the values one through seven.

Before you begin:

Add a question of the slider type to your survey. For more information, see Adding a question to a question page.

©2025 Pegasystems Inc

> **NOTE:** Only traditional UI Pega Platform applications support questions with a slider. Applications based on Constellation do not display questions with sliders.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	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 list of Case Types, open a Case Type of the `Questionnaire` type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Survey**. b. Click the questionnaire that you want to edit.

2. If you have multiple questions on the question page, click the question that you want to edit.
3. From the list of question types, select **Slider**.
4. In the rich text editor, enter your question.

> **For example:**
>
> Enter `How many days a week do you work?`.

5. In the **Question settings** pane, format the slider:
   a. In the **Starting range** field, enter the minimum numerical value for the scale, for example, enter `1`.

©2025 Pegasystems Inc

b. In the **Ending range** field, enter the maximum numerical value for the scale, for example, enter 7.
c. In the **Interval** field, enter the numerical interval between the possible values on the scale, for example, enter 1.
d. **Optional:** To preset the answer to the question, enter a value in the **Default value** field, for example, enter 5.
e. **Optional:** To show possible values on the slider, select the **Show captions** checkbox.

6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a slider with captions at run time:

Slider with captions

Related tasks

- Designing questionnaires
- Fine-tuning your questionnaire

Related reference

©2025 Pegasystems Inc

- Best practices for creating questionnaires

Formatting a radio button matrix in a questionnaire

Format a radio button matrix in a survey to ask a series of related questions that users answer by selecting one option from a repeating group of radio buttons. In a radio button matrix, in each row, users can select an answer from only one column. For example, you can ask users to rate their pain threshold on a scale of one to five for different activities, such as swimming, running, or walking.

Before you begin:

Add a question of the radio button matrix type to your questionnaire. For more information, see Adding a question to a question page.

NOTE: Only traditional UI Pega Platform applications support questions with a radio button matrix. Applications based on Constellation do not display questions with a radio button matrix.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	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 list of Case Types, open a Case Type of the `Questionnaire` type that you want to edit.

©2025 Pegasystems Inc

Choices	Actions
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click Configure > Case Management > Survey. b. Click the questionnaire that you want to edit.

2. If you have multiple questions in the question page, click the question that you want to edit.
3. In the list of question types, select Radio button matrix.
4. In the rich text editor, enter your question.

   For example:

   Enter `Rate your pain threshold during different activities`.

5. In the **Columns** section, create columns for the matrix:
   a. Click Add column.
   b. In the text field, enter a label for the column.

      For example:

      Enter phrases to describe different levels of pain, such as `No pain` or `Moderate pain`.

   c. **Optional:** To add more columns, repeat steps 5.a through 5.b.
6. In the **Rows** section, create rows for the matrix:
   a. Click Add row.
   b. In the text field, enter a label for the row.

      For example:

      Enter different activities, such as `Swimming` or `Running`.

©2025 Pegasystems Inc

7. **Optional:** To store the selected value from a column under a name that is different from the label, change the reporting value:
   a. Click **More** > **Advanced view**.
   b. In the **Reporting value** field, replace the default value with your custom value.

   > **For example:**
   >
   > You can label the columns with adjectives describing different levels of pain, and then store the results as numbers.

8. **Optional:** To display a row only in selected scenarios, configure Visibility Conditions:
   a. Click **More** > **Advanced view**.
   b. From the **Visible when** list, select **When**, and then click **Edit**.
   c. Provide a value to compare at run time, a comparator, and a value to compare against the first value.

   > **For example:**
   >
   > To display `Swimming` only to users who declare performing this activity, create the Condition `Activity is in list "Swimming", "Running", "Walking"`.

   d. **Optional:** To add more Conditions, click **Add a row**.
   e. If you create multiple rows, define whether an answer needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
   f. Click **Submit**.
9. Click **Save**.
10. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

©2025 Pegasystems Inc

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a radio button matrix to rate pain threshold during different activities:

Rate your pain threshold during different activities

	No pain	Mild pain	Moderate pain	Strong pain	Severe pain
Swimming	○	○	○	○	○
Running	○	○	○	○	○
Walking	○	○	○	○	○
Resting	○	○	○	○	○

Radio button matrix at run time

What to do next:

Provide tools to compare and calculate weight of responses by scoring answers in the radio button matrix. For more information, see Scoring answers in a questionnaire.

- Designing questionnaires
- Fine-tuning your questionnaire

Formatting a currency question in a questionnaire

Enable users of your application to provide answers that precisely fit their financial business scenarios by creating a question of a currency type. As a result, users can clearly link the number that they provide to the amount of money in a specified currency. For example, if a customer has an international account, by providing a currency in a question about a daily transactions limit, you help users understand how their answers affect their bank account.

©2025 Pegasystems Inc

Before you begin:

Add a question of a currency type to a question page. For more information, see Adding a question to a question page.

NOTE: Only applications that you build on Constellation support currency questions.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Currency**.
4. In the **Question settings** pane, provide the currency that you want to use:

Actions	Choices
Enter the currency manually	a. In the **ISO code selection** list, select **Constant**.

©2025 Pegasystems Inc

Actions	Choices
	b. In the **Currency ISO code**, enter the currency code. **For example:** To display a question that captures the amount of US dollars, enter `USD`.
Source the currency from a property	a. In the **ISO code selection** list, select **Property Reference**. b. In the **Currency ISO code** list, select a property that stores the currency that you want to use.

5. To enable users to provide more precise answers, select the **Allow decimals** checkbox.

   Result:

   At run time, users can provide answers that include decimals.

6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

For example:

The following figure shows a sample currency question about a daily limit on financial transactions with an answer that includes a decimal:

A currency question

- Making a question required in a questionnaire
- Conditionally displaying a question in a questionnaire
- Conditionally displaying a question page in a questionnaire

Formatting a decimal question in a questionnaire

Help users of your applications provide detailed and accurate answers that require numbers by creating decimal questions. At run time, users can enter numbers that include decimals after a separator to provide very detailed information. As a result, your application can precisely meet even the most specific business requirements. For example, in a questionnaire about the reimbursement of travel costs, users can provide the precise amounts of their travel expenses.

Before you begin:

Add a question of a decimal type to a question page. For more information, see Adding a question to a question page.

NOTE: Only applications that you build on Constellation support decimal questions.

1. Navigate to the questionnaire:

©2025 Pegasystems Inc

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the list of question types, select **Decimal**.
4. In the **Question settings** pane, in the **Decimal precision** field, enter an integer that reflects how many places after the decimal separator you want users to be able to enter figures.

   For example:

   To enable users to provide a precise amount of money that includes dollars and then cents, enter `2`.

   Result:

   At run time, users can provide numbers such as `25.55`.

5. For greater clarity, to display a separator in the number, select the **Show thousands separator** checkbox.
6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

©2025 Pegasystems Inc

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a decimal question about costs of accommodation with an answer that includes a thousands separator and a two digit decimal separator:

A decimal question at run time

- Formatting a currency question in a questionnaire
- Making a question required in a questionnaire
- Conditionally displaying a question in a questionnaire
- Conditionally displaying a question page in a questionnaire
- Validating answers in a questionnaire

Scoring answers in a questionnaire

Calculate a cumulative score based on your questionnaire results, to generate metrics and trends for a specific focus group. For example, in a questionnaire about satisfaction with customer service, you can assign high scores to better ratings and low scores to poorer ratings to quickly calculate the general satisfaction level of customers.

> **NOTE:** Only traditional UI Pega Platform applications support scoring answers. You cannot configure scoring values in applications based on Constellation.

©2025 Pegasystems Inc

You can configure scoring only for questions that you add to a question page in a questionnaire. Individual questions that you ask by using the Question shape in a Case Life Cycle do not support scores.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. Click a question with one of the following answer formats:
   - Picklist
   - Checkbox
   - Radio button matrix
3. In the **Question settings** pane, in the **Scoring** section, click the **Properties** icon.
4. In the **Binary** window, assign numeric values to the choices.

©2025 Pegasystems Inc

For example:

Configuring scores for answers

5. Click **Submit**.
6. Click **Save**.
7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

When users provide answers, your application calculates the cumulative score for the questionnaire, and saves the score in the *pxCumulativeQuestionScore* property.

- Formatting a picklist in a questionnaire
- Formatting a radio button matrix in a questionnaire

©2025 Pegasystems Inc

- Formatting a checklist in a questionnaire

Best practices for creating questionnaires

You can use questionnaires to gather information from users. Apply best practices that can help you create questionnaires that are easy to use and maintain. For example, if you create a questionnaire about customer service satisfaction with short, precise questions grouped into pages based on the service area, you can then quickly adjust the visibility of particular pages based on the user input, and conveniently reuse parts of your questionnaire in the future.

To ensure that your questionnaires achieve the best results, apply the following best practices:

- Ensure that users stay focused and can provide answers efficiently by grouping questions in a logical way.

  For example, in a questionnaire that collects information about customer satisfaction with a hotel, ask several questions about the service quality, and then provide a series of questions about the food.

- Limit the total number of questions. You can also divide a questionnaire into smaller questionnaires that your application calls in succession.

  Users tend to provide less reliable and specific answers in longer questionnaires.

- Limit the number of questions on each page.

  Many questions might cause excessive scrolling.

- Use concise questions.

  Users are more likely to provide accurate answers to short, precise questions.

- Ensure that the format of the question matches the type of information that you want to gather.

PEGA ©2025 Pegasystems Inc

For example, if you want users to select only one option that describes their favorite type of service that a company offers, create a picklist instead of multiple checkboxes. In another example, if you want users to rate their level of satisfaction with a service, create a slider instead of an answer with multiple checkboxes.

- Proofread your questionnaire.

  Grammatical errors, duplicate questions, and acronyms or terms that you do not explain might discourage users from providing useful answers.

- Ask potential users for feedback on the clarity, length, and ease of use of your questionnaire.
- Ensure that your questionnaires are flexible and relevant in multiple scenarios by applying Visibility Conditions.

  For more information, see Conditionally displaying a question in a questionnaire.

- Ensure that users provide answers in the correct format by using Validations.

  For example, you can process cases faster by ensuring that users can enter only past dates of stay at a hotel. For more information, see Validating answers in a questionnaire.

- Designing questionnaires
- Creating a questionnaire
- Running a questionnaire in a Case

Running a questionnaire in a Case

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

PEGA ©2025 Pegasystems Inc

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

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

Before you begin:

- Create a questionnaire that you want to run in a Case. For more information, see Creating a questionnaire.
- If you update your questionnaires from Pega Platform 8.5 or earlier and you want to access questionnaire data from the parent Case, first remove and then reconfigure the **Questionnaire** Step in the parent Case, and next edit the *DataTableEditorReport* report definition by adding the *parentKey* parameter to a filter condition. Ensure that the *parentKey* parameter equals to the *.pxCoverInsKey* column source. For more information, see Report definitions.

1. Add the **Questionnaire** Step to the lifecycle of your Case.
   a. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
   b. On the **Workflow** tab, in the **Case Life Cycle** section, hover over a Process, and then click **Step** > **More** > **Automations** > **Questionnaire**.

©2025 Pegasystems Inc

> **NOTE:** If the Process in which you want to add the Step includes advanced shapes, such as a Decision shape, click **Configure Process**, and then finalize the configuration by adding and connecting Process shapes. For more information, see step 2 in Creating a stand-alone Process and Process shapes.

   c. Click **Select**.

2. In the **Create Questionnaire** field, provide a name for the Step, for example, `Rate customer service`.
3. In the **Step** properties pane, in the **Select questionnaire** list, select the name of the questionnaire that you want to run as a child Case.
4. **Optional:** To reuse information from the parent Case, transfer data from the current Case to the questionnaire:

Choices	Actions
Map fields from the Case Type to the questionnaire	a. In the **Step** properties pane, select the **Transfer info to questionnaire Case** checkbox. b. In the **Transfer information** dialog box, in the **'From' field** column, select the field with the values that you want to transfer. c. In the **'To' field** column, press the Down arrow key, and then select a destination field to which you want to transfer the value. 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** checkbox.

©2025 Pegasystems Inc

Choices	Actions
	e. Click **OK**.
Configure advanced mapping by providing a Data Transform	a. Switch to Dev Studio. b. In the **Step** properties pane, select the **Transfer info to questionnaire Case** checkbox. c. In the list of transfer options, select **Existing**. d. In the **Data Transform** field, enter the name of the Data Transform that you want to apply. e. Click **Save**, and then switch to App Studio.

Result:

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

5. In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the questions in the questionnaire:
   - To route the questionnaire to the user who currently processes the Case, select **Current user**.
   - To route the questionnaire to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to complete the questionnaire.
   - To route the questionnaire to a Work Queue that multiple users share, select **Work Queue**, and then, in the **Work Queue** field, enter the name of the Work Queue.
6. In the upper-right corner of the Case Type, save your changes by clicking **Save**.

©2025 Pegasystems Inc

7. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

After you add a Questionnaire Step to your parent Case Type, you can access the data that users provide only after the child Case for completing the questionnaire is complete. In the Data Model of the parent Case, the system adds two new properties of a query type: a single property, and a list property. These two new properties source data from the questionnaire child Case Type. For more information about queries, see Querying a Data Page.

For example:

The following figure shows the effect of propagating data from a parent Case to a questionnaire child Case at run time in a process of reviving a reimbursement claim in an application for hotel operations. After a customer provides details, such as dates of arrival and departure, in the parent Case, when the questionnaire runs as a child Case, the application prepopulates answers with information that the customer already provided:

PEGA ©2025 Pegasystems Inc

Prepopulating answers in a questionnaire

- Designing questionnaires
- Sharing data between parent and child Cases
- Fine-tuning your questionnaire

Fine-tuning your questionnaire

Refine the way that your questionnaire displays questions and evaluates answers to ensure that you collect valid information in a Case. For example, you can define conditions for displaying a question or a question page, validate the answers that the users provide, and define the run-time order of the questions in your questionnaire.

Use the following techniques to fine-tune your questionnaire:

- **Making a question required in a questionnaire**

©2025 Pegasystems Inc

- **Conditionally displaying a question in a questionnaire**
- **Conditionally displaying a question page in a questionnaire**
- **Displaying questions in a read-only mode**
- **Validating answers in a questionnaire**
- **Displaying a warning in a questionnaire**
- **Changing the run-time order of questions in a questionnaire**
- **Displaying a block of text in a question page**

- Designing questionnaires
- Changing the run-time order of questions in a questionnaire

Making a question required in a questionnaire

Make a question required to ensure that your questionnaire collects complete information. For example, in a questionnaire about the service at a hotel, if a customer rates service as poor, then a field for additional comments is required.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.

©2025 Pegasystems Inc

Choices	Actions
	b. Click the questionnaire that you want to edit.

2. Click the question that you want to make required.
3. In the **Question settings** pane, in the **Required When Conditions** section, click the Properties icon.
4. In the **Conditions** dialog box, create a Condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

   For example:

   To make the question required for VIP customers, create the Condition `CustomerStatus equals to vIP`.

5. **Optional:** To add more Conditions, click Add a row.
6. If you create multiple Conditions, define whether a question needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
7. Click OK.
8. Save your changes by clicking Save.
9. **Optional:** To view how your changes are reflected at run time, click Preview and run the Case.

IMPORTANT: Avoid using Save and run to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

Result:

At run time, if the When Conditions evaluate to true, the question is required to submit the questionnaire.

©2025 Pegasystems Inc

- Validating answers in a questionnaire
- Defining Conditions in the Condition builder

Conditionally displaying a question in a questionnaire

Define Conditions to ensure that your questionnaire displays relevant questions, based on the context of the questionnaire participant. For example, in an onboarding application, a question about a preferred office chair is displayed only if the employee does not work remotely.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. Click the question for which you want to configure Visibility Conditions.
3. In the **Question settings** pane, in the **Visibility Conditions** section, click the **Properties** icon.
4. In the **Conditions** dialog box, create a Condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

> **For example:**
>
> To show the question only to users who have a Gold or Platinum account, create a Condition `AccountType is in list "Gold", "Platinum"`.

©2025 Pegasystems Inc

5. **Optional:** To add more Conditions, click **Add a row**.
6. If you create multiple Conditions, define whether a question needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
7. Click **OK**.
8. Save your changes by clicking **Save**.
9. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **Result:**
>
> At run time, the questionnaire displays the question only if the Visibility Conditions evaluate to true.

- Conditionally displaying a question page in a questionnaire
- Defining Conditions in the Condition builder

Conditionally displaying a question page in a questionnaire

Define Conditions to ensure that your questionnaire displays related questions that are relevant to a given scenario, based on the context of the questionnaire participant. For example, in an onboarding application, the questionnaire displays questions about preferred office accommodations only if the employee does not work remotely.

1. Navigate to the questionnaire:

©2025 Pegasystems Inc

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. Click the question page for which you want to configure Visibility Conditions.
3. In the **Page settings** pane, in the **Visibility Conditions** section, click the **Properties** icon.
4. In the **Conditions** dialog box, define a Condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

   > **For example:**
   >
   > To display the question page only to customers who stay at a hotel for five days or more, enter `Duration is greater than or equal to 5`.

5. **Optional:** To add more Conditions, click **Add a row**.
6. If you create multiple Conditions, define whether a Case needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
7. Click **OK**.
8. Save your changes by clicking **Save**.
9. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

©2025 Pegasystems Inc

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **Result:**
>
> At run time, the question page is displayed only if the Visibility Conditions evaluate to true.

- Conditionally displaying a question in a questionnaire
- Defining Conditions in the Condition builder

Displaying questions in a read-only mode

Help users provide accurate answers and speed up the time required to complete a questionnaire by displaying questions in a read-only mode. The read-only mode of questions automates the completion of questionnaires when you reference data for an answer from another questionnaire or question, or when the answer from one question directly relates to other questions.

For example, you can create a questionnaire that captures customer's preferences about their bank account. If the customer already provided their monthly income in another Case or questionnaire, the application can display a question about their monthly income in the bank account questionnaire in a read-only mode.

> **NOTE:** Only applications that you build on Constellation support read-only mode for questions. Questions of the info text and file types do not support read-only mode.

To display a question in a read-only mode, you define a Condition in a Condition Builder. At run time, if the Condition evaluates to true, your application displays the

©2025 Pegasystems Inc

question in a read-only mode and prepopulates the answer with the value that the user provided in another Case in the field that you use in the Condition Builder.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. If you have multiple questions on the question page, click the question that you want to edit.
3. In the **Question settings** pane, in the **Read-only When Conditions** section, click the **Configure** icon.
4. In the **Conditions** dialog box, create a Condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

> **For example:**
>
> To display the question in a read-only mode when a user provided their monthly income in another questionnaire, enter `What is your monthly income? is greater than 0`, as in the following example:

©2025 Pegasystems Inc

Conditions for a read-only question mode

Result:

At run time, if the Condition evaluates to true, a questionnaire displays the value that a user provided as an answer to the question `What is your monthly income?` in another questionnaire.

5. **Optional:** To add more Conditions, click **Add a row**.
6. If you create multiple Conditions, define whether a question needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
7. Click **OK**.
8. Save your changes by clicking **Save**.
9. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Defining Conditions in the Condition builder
- Conditionally displaying a question in a questionnaire
- Conditionally displaying a question page in a questionnaire

Validating answers in a questionnaire

Display an error message in your questionnaire when an answer is invalid, to ensure that the information that users provide is accurate and in an expected format. For

©2025 Pegasystems Inc

example, if a user types letters in a phone number, your application prompts the user to enter only numbers in that field.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**. b. Click the questionnaire that you want to edit.

2. Click the question for which you want to configure a Validation.
3. In the **Question settings** pane, in the **Validations** section, click the **Properties** icon.
4. In the **Validate fields based on these Conditions** dialog box, in the **Message** field, enter an error message that users receive if the Validation fails.

   For example:

   Enter `Please correct your date of birth.`

5. Create a Condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

   For example:

   To ensure that users enter a date of birth in the past, create a Condition that checks the date. For example, `BirthDate is before` *1/22/2021*.

©2025 Pegasystems Inc

6. **Optional:** To add more rows to the Condition, click **Add a row**.
7. If you create multiple rows, define whether an answer needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
8. **Optional:** To add more Validations, click **Add Condition**, and then repeat steps 4 through 7.
9. Click **OK**.
10. Save your changes by clicking **Save**.
11. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

> **Result:**
>
> If the Validation fails at run time, users receive an error message. Users need to correct the answers with failed Validation before submitting the questionnaire.

- Making a question required in a questionnaire
- Displaying a warning in a questionnaire
- Populating a question template
- Defining Conditions in the Condition builder

Displaying a warning in a questionnaire

Use a warning message to encourage questionnaire participants to provide complete answers. For example, if a user enters their birth date in the future, your application displays a reminder that the date must be in the past and be consistent with the customer being an adult.

1. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.

©2025 Pegasystems Inc

2. On the Questions tab, click a question that you want to edit.
3. Expand the **Warnings** section.
4. Define the Conditions that control when your application displays the warning.

Choices	Actions
Use a Boolean Expression **NOTE: Use this approach when users complete your questionnaire offline.**	a. In the Visibility list, select Condition (Expression). b. In the Warning Expression field, enter a string Expression that evaluates property values in the questionnaire.
Use a When Condition **NOTE: Use this approach when users complete your questionnaire online.**	a. In the Visibility list, select Condition (When Rule). b. In the Warning when field, enter the name of a When Condition that evaluates property values in the questionnaire.
Build a custom Condition that can be a combination of When Conditions and Boolean Expressions	a. In the Visibility list, select Condition (Expression). b. Click the Open Condition Builder icon. c. In the Visible when dialog box, create a Condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

©2025 Pegasystems Inc

Choices	Actions
	For example: To show the warning only to users who have a Gold or Platinum account, create a Condition `AccountType is in list "Gold", "Platinum"`. d. **Optional:** To use two Conditions, define whether a question needs to meet both or any of the Conditions by selecting either **and** or **or** between the rows. e. Click **Save**.

5. In the **Warning message** field, enter the name of a Rule that contains a warning message.
   For more information, see Messages.
6. Save your changes by clicking **Save**.
7. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Validating answers in a questionnaire

©2025 Pegasystems Inc

Changing the run-time order of questions in a questionnaire

Conditionally skip or revisit questions at run time by adding a branch to a questionnaire. By branching questionnaires, you provide flexibility and promote reuse of existing assets, such as question pages.

For example, in a shopping application, you can define a Condition that skips a set of questions directed towards new customers if a customer reports making purchases in the past. If the customer indicates an interest in providing feedback about their previous purchases, the questionnaire displays questions about their satisfaction with the service. For a new customer, your application loads a set of questions about how the customer found out about your company.

Before you begin:

- Create a questionnaire. For more information, see Creating a questionnaire.
- Create a Data Model for your questionnaire. For more information, see Extending a Data Model for a Case.

1. Navigate to the questionnaire:

Choices	Actions
Open the questionnaire in App Studio	a. In the navigation pane of App Studio, click **Case Types**. b. In the list of Case Types, open a Case Type of the Questionnaire type that you want to edit.
Open the questionnaire in Dev Studio	a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.

©2025 Pegasystems Inc

Choices	Actions
	b. Click the questionnaire that you want to edit.

2. On the **Questionnaire** tab, click **Add** > **Logic**.
3. In the **Branch** field, click **Click here to edit page title**, and then enter a unique name for the branch, for example `Returning customer`.
4. In the **Branching Conditions** pane, in the **When** section, click **Click to add Condition**.
5. Create a Condition by providing a value to compare at run time, a comparator, and a value to compare against the first value.

   For example:

   To skip a question if the customer has VIP status, create the following Condition: `IsCustomerVIP is true`.

6. **Optional:** To add more rows in the Condition, click **Add a row**.
7. If you create multiple rows, define whether an answer needs to meet all or any Conditions by selecting either **and** or **or** between the rows.
8. Click **Submit**.
9. In the **Go to** list, define what happens when the Condition evaluates to true:
   - To move to a specific questionnaire item, select the item in the list.
   - To exit the questionnaire, select **Exit Survey**.

   For example:

   You can skip to the next question page in a personal health questionnaire, when users indicate that they do not smoke.

10. **Optional:** To define more Conditions, click **Add path**, and then repeat steps 4 through 9.
11. In the **Otherwise go to** list, define what happens when the Condition returns a false value:

©2025 Pegasystems Inc

- To move to a specific questionnaire item, select the item in the list.
- To exit the questionnaire, select **Exit Survey**.

12. Save your changes by clicking **Save**.
13. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> ⚠ **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

- Designing questionnaires
- Creating a questionnaire

Displaying a block of text in a question page

Provide supporting information that helps users answer questions more quickly by displaying a block of read-only text on a question page of your questionnaire. For example, you can add a paragraph to explain to users that a complete answer to the next question helps you improve your application.

> **Before you begin:**
>
> Add a question page to your questionnaire. For more information, see Adding a question page to a questionnaire.

1. In the navigation pane of App Studio, click **Case Types**.
2. In the list of Case Types, open a Case Type of the `Questionnaire` type that you want to edit.
3. On the **Questionnaire** tab, on the question page that you want to edit, add a block of text:

©2025 Pegasystems Inc

Choices	Actions
Create a new block of text	a. Click Add question > New. b. In the list of question types, select Info text. c. In the text editor, enter informative text. **For example:** Enter `All questionnaire data is confidential. We use the information that you give us only for statistical purposes and do not share it with any third parties.`
Reuse an existing block of text	a. Click Add question > From library. b. In the **Questionnaire library** dialog box, select a block of text that you want to reuse, and then click Add next to the text. c. Close the dialog box by clicking Add.

NOTE: In applications based on Constellation, you can use only plain text info text. Traditional UI Pega Platform applications support inserting dynamic values and formatting the question text. For more information, see Application UI setup and Center-out configuration in Constellation.

4. Save your changes by clicking Save.

©2025 Pegasystems Inc

5. **Optional:** To view how your changes are reflected at run time, click **Preview** and run the Case.

> **IMPORTANT:** Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

For example:

The following figure shows a block of text that provides additional information about the questionnaire:

A block of text in a question page at run time

What to do next:

Add more questions to your questionnaire. For more information, see Adding a question to a question page.

- Fine-tuning your questionnaire

©2025 Pegasystems Inc

- Adding a question to a question page

Visualizing questionnaire data by creating Insights in Constellation

Analyze and manage data from questionnaires in your application by using Insights. For example, you can view the service quality questionnaires from multiple respondents at once, and display their responses as an interactive chart that provides an overview of their satisfaction with the service.

> **Before you begin:**
>
> Create a questionnaire in your application. For more information, see Creating a questionnaire.

Because a questionnaire is a regular Case Type, you create and use Insights for questionnaires in the same way as for any other Case Type.

1. In your application, create an Insight for your questionnaire Case Type.

   > **For example:**
   >
   > In a banking application, create an Insight for a Customer Satisfaction questionnaire, which gathers the users' satisfaction with their interactions with the bank.

   For more information, see Creating Insights.
2. **Optional:** To adjust the data display to your individual needs, customize how the Insight presents data from your questionnaire.

   > **For example:**
   >
   > Filter an Insight to include only the questionnaires in which the respondents indicated a monthly income of at least $20,000, so that you can analyze the satisfaction of VIP customers. Then, you can group the results by the application components that the customers use, to decide

©2025 Pegasystems Inc

> which components of your application need improvement to satisfy the most valuable customers.

For more information, see Customizing table-based Insights

3. **Optional:** To visualize data in a user-friendly way that helps you analyze the answers, create a chart that presents answers from your questionnaire.

> **For example:**
>
> Visualize hundreds of responses from a customer satisfaction questionnaire as a pie chart, and save the chart to conveniently monitor the current results every time you run the application.

For more information, see Customizing chart-based Insights.

- Creating a questionnaire
- Insights in Case Management
- Creating Insights

Using advanced questionnaire features in Dev Studio

Use Dev Studio to collect information from users in advanced use cases for which providing questionnaires in a low-code way in App Studio is not sufficient. For example, you can run a single question in a Case Lifecycle to use the question in offline mode, reuse answers in other questions, and import a large number of questions by using an Excel template.

You can complete the following tasks in Dev Studio:

- **Creating a stand-alone question with a single answer**
- **Creating a stand-alone question with multiple answers**
- **Reusing answers in questions**
- **Creating questions in bulk**

PEGA ©2025 Pegasystems Inc

- **Asking a question in a Case**
- **Asking a group of related questions in a Case**
- **Creating reports based on questionnaires in Dev Studio**
- Extending a Data Model for a Case
- Running a questionnaire in a Case
- Best practices for creating questionnaires

Creating a stand-alone question with a single answer

Support information gathering when a Case is offline by creating a stand-alone, single-answer question, which you can reference outside the context of a questionnaire. For example, during a conversation with a customer service representative (CSR), you can ask the customer a single question about their satisfaction with the CSR's service.

1. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.
2. Click the **Questions** tab.
3. Click **Create question** > **Browse all types** > **Single answer** > *[answer type]*.
4. In the **Name** field, enter a unique name that describes the purpose of the question, for example, enter `Employee address`.
5. **Optional:** To change the Scope of the question, expand the **Context** section and provide a different class name, Ruleset name, or Ruleset version.
   The Scope controls which Rules can reference your question and which Rules your question can reference.
6. **Optional:** To manage your questions efficiently, in the **Category** field, you can enter a category for the question, for example, enter `CustomerService`.
7. In the **Question layout** list, select how your application sources the text of the question:

Choices	Actions
Plain text that you provide	a. Select **Plain Text**.

©2025 Pegasystems Inc

Choices	Actions
	b. In the text editor, enter the text of the question.
Rich text that you provide **NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option.**	a. Select Rich text. b. In the **Question text** field, enter your question. c. **Optional:** To make the output of the question more accurate, apply formatting to your text. **For example:** Emphasize certain words with the bold style, and apply a numbered list.
Custom HTML Rule **NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option.** **NOTE: Use a custom HTML Rule only in specific use cases that you cannot complete by reusing the standard Rules.**	a. Select Custom HTML. b. In the **Custom HTML** field, enter the name of the Rule that contains a custom HTML question.
A Section Rule in your application	a. Select Section.

©2025 Pegasystems Inc

Choices	Actions
NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option.	b. In the **Section reference** field, enter the name of the Rule that contains a Section that you want to reuse.
A Paragraph Rule in your application **NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option.**	a. Select Paragraph. b. In the **Section reference** field, enter the name of the Rule that contains a Section that you want to reuse.

8. If you provide a text box for the answer, in the **Answer presentation settings** section, define how your questionnaire displays the text box:
   a. In the **Default** field, enter the default text of the answer.
   b. In the Size of text box field, enter the maximum length for the answer, measured in characters and spaces.
9. If you provide a date picker for the answer, in the **Answer presentation settings** section, in the Default field, enter the default date.
10. If you provide a date-time picker for the answer, in the **Answer presentation settings** section, in the Default field, enter the default date and time.
11. If you provide a text area for a long answer or a comment, in the **Answer presentation settings** section, define how your questionnaire displays the long answer:
    a. In the Default field, enter the default text of the answer.
    b. In the Rows field, enter the height of the text area, measured in characters.

PEGA ©2025 Pegasystems Inc

   c. In the **Columns** field, enter the width of the text area.
12. If you provide a custom drop-down list for the answer, in the **Drop down type** list, select **Prompt list**, and then provide choices:
    a. **Optional:** In the **When** column, enter a When Condition that controls whether your application displays the choice.
    b. In the **Display value** column, enter a label for the choice.
    c. In the **Score** column, enter a score value that your application stores when the user selects the response.
    d. **Optional:** To select an answer by default, in the **selected** column, select the radio button.
    e. **Optional:** To add another answer, click **Add a row**.
13. If you provide a predefined drop-down list for the answer, in the **Drop down type** list, select **Data page**, and then provide choices:
    a. In the **Data page** field, enter a Data Page that returns a list of options.
    b. **Optional:** To pass parameters that the Data Page supports, in the **Value** field, enter the parameters.
    c. In the **Property for value** field, enter a property on the Data Page that stores the options for the list.
    d. **Optional:** To customize the labels of the choices, in the **Property for display text** field, enter a property on the Data Page that contains the labels.
14. If you provide checkboxes for the answer, in the **Answer presentation settings** section, in the **Score** field, enter a score value that your application stores when the user selects the checkbox.
15. If you provide custom radio buttons for the answer, in the **Radio button source** list, select **Prompt list**, and then provide choices:
    a. In the **When** column, enter a When Condition that controls whether your application displays the choice.
    b. In the **Display Value** column, enter a label for the choice.
    c. In the **Score** column, enter a score value that your application stores when the user selects the response.

PEGA ©2025 Pegasystems Inc

   d. **Optional:** To select an answer by default, in the **selected** column, select the radio button.
   e. **Optional:** To add another answer, click **Add a row**.
16. If you provide predefined radio buttons for the answer, in the **Radio button source** list, select **Data page**, and then provide choices:
   a. In the **Data page** field, enter a Data Page that returns a list of options.
   b. **Optional:** To pass parameters that the Data Page supports, in the **Value** field, enter the parameters.
   c. In the **Property for value** field, enter a property on the Data Page that stores the options for the list.
   d. **Optional:** To customize the labels of the choices, in the **Property for display text** field, enter a property on the Data Page that contains the labels.
17. If you provide a slider for the answer, in the **Answer** section, define the parameters of the slider:
   a. In the **Starting range** field, enter the minimum numerical value for the scale, for example, enter 1.
   b. In the **Ending range** field, enter the maximum numerical value for the scale, for example, enter 7.
   c. In the **Interval** field, enter the numerical interval between the possible values on the scale, for example, enter 1.
   d. **Optional:** To preset the answer to the question, enter a value in the **Default value** field, for example, enter 5.
   e. **Optional:** To show possible values on the slider, select the **Show captions** checkbox.
18. If you provide a custom section for the answer, in the **Answer presentation settings** section, in the **Section reference** field, enter a section that displays fields for the user to enter a response.
19. In the **Map to property** list, select a property in the Data Model that stores the answer.

PEGA ©2025 Pegasystems Inc

> **NOTE:** Some types of questions do not support property mapping, for example, info text.

20. **Optional:** To ensure that the information that users provide is accurate and in an expected format by displaying an error message in your questionnaire when the answer is invalid, in the **Answer** section, in the **Answer Validation** field, enter the name of a Validation Rule.
21. **Optional:** To encourage questionnaire participants to provide complete answers, configure a warning. For more information, see Displaying a warning in a questionnaire.

> **What to do next:**
>
> Organize your stand-alone question with related questions by adding it to a question page. For more information, see Adding a question to a question page.

- Asking a question in a Case
- Working with offline-enabled mobile apps
- Creating a stand-alone question with multiple answers
- Creating questions in bulk

Creating a stand-alone question with multiple answers

Gather information when a Case is offline by creating a stand-alone, multi-answer question in Dev Studio. You can reference stand-alone questions outside the context of a questionnaire, for example, while processing an order, you can ask the customer a single question about their satisfaction with the sales application.

> **NOTE:** Only traditional UI Pega Platform applications support questions with multiple answers. Applications based on Constellation display only single-answer questions.

©2025 Pegasystems Inc

1. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.
2. Click the **Questions** tab.
3. Click **Create question** > **Browse all types** > **Multi answer** > *[answer type]*.
4. In the **Name** field, enter a unique name that describes the purpose of the question, for example, enter `Customer service rating`.
5. **Optional:** To change the Scope of the question, expand the **Context** section and provide a different class name, Ruleset name, or Ruleset version.
   The Scope controls which Rules can reference your question and which Rules your question can reference.
6. **Optional:** To manage your questions efficiently, in the **Category** field, you can enter a category for the question, for example, enter `SatisfactionRatings`.
7. In the **Question layout** list, select how your application sources the text of the question:

Choices	Actions
Plain text that you provide	a. Select **Plain Text**. b. In the text editor, enter the text of the question.
Rich text that you provide **NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option.**	a. Select **Rich text**. b. In the **Question text** field, enter your question. c. **Optional:** To make the output of the question more accurate, apply formatting to your text. **For example:** Emphasize certain words with the bold style, and apply a numbered list.
Custom HTML Rule	a. Select **Custom HTML**.

©2025 Pegasystems Inc

Choices	Actions
NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option. **NOTE: Use a custom HTML Rule only in specific use cases that you cannot complete by reusing the standard Rules.**	b. In the **Custom HTML** field, enter the name of the Rule that contains a custom HTML question.
A Section Rule in your application **NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option.**	a. Select Section. b. In the **Section reference** field, enter the name of the Rule that contains a Section that you want to reuse.
A Paragraph Rule in your application **NOTE: Available only in traditional UI Pega Platform applications. Applications based on Constellation do not support this option.**	a. Select Paragraph. b. In the **Section reference** field, enter the name of the Rule that contains a Section that you want to reuse.

©2025 Pegasystems Inc

8. **Optional:** To format the answers, in the **Answer** section, define how your questionnaire displays the available choices in the question.

   > **For example:**
   >
   > Format a table by providing a class that contains a table, adding columns that reference column properties, and adding several rows.

9. Define how your application stores the answer in the Data Model:
   - To not map the answer, in the **Property type** list, select **No Property Mapping**.
   - To store the answer as a single-value property, in the **Property type** list, select **Single Value**.
   - To store the answer as a page list, in the **Property type** list, select **Page List**, and then, in the **Page list** field, enter a page list property to store the answer.

   > **NOTE:** The **Property type** list is not available for questions of the table type, because you configure property mapping while configuring each column by completing the **Column property** field.

10. **Optional:** To ensure that the information that users provide is accurate and in an expected format by displaying an error message in your questionnaire when the answer is invalid, in the **Answer** section, in the **Answer Validation** field, enter the name of a Validation Rule.

    > **NOTE:** Answer Validation is not available for questions of the table type.

11. **Optional:** To encourage questionnaire participants to provide complete answers, configure a warning. For more information, see Displaying a warning in a questionnaire.

PEGA ©2025 Pegasystems Inc

Related tasks

- Creating a stand-alone question with a single answer
- Asking a question in a Case
- Working with offline-enabled mobile apps

Reusing answers in questions

Create more personalized and comprehensive questionnaires by reusing the answers to questions in subsequent questions. For example, in a sales application, a question prompts the user to enter their name, and then all following questions can start with the answer that the user provides. You can reuse only answers to questions that already exist in your questionnaire or in the base class in which you create your questionnaire.

> **Before you begin:**
>
> Create a questionnaire in which you want to reuse answers. For more information, see Creating a questionnaire.

> **NOTE:** Only traditional UI Pega Platform applications support reusing answers in questions. Applications based on Constellation do not support this option.

1. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.
2. Click the questionnaire that you want to edit.
3. Click **Add question** > **New**.
4. In the list of question types, select the type of question that you want to create. For more information about creating questions, see Adding a question to a question page.
5. In the text field, enter a question, and then place the cursor where you want to reuse the answer.

©2025 Pegasystems Inc

6. On the text editor toolbar, click the **Insert Property** icon.
7. In the **Select from question response** dialog box, select a question from which you want to reuse the answer.

For example:

Reusing an existing answer in a block of text

The question that you select must appear in the questionnaire before the question that you are creating.

8. Click **Save**.
9. **Optional:** To view your changes, switch to App Studio, and then click **Preview** and run the Case.

IMPORTANT: Avoid using **Save and run** to test your changes because frequent saves on a Case Type might overwrite work of other developers even if you did not perform any changes and intend to just run a Case.

©2025 Pegasystems Inc

Result:

At run time, your application populates a property with a value that users provided earlier in the questionnaire.

For example:

Reusing an existing answer in another question to address a user by name

- Adding a question to a question page

Creating questions in bulk

Develop your questionnaires more quickly by creating multiple questions at the same time. For example, a marketing specialist creates a large number of questions and their variations to collect information for a new campaign, and an application developer uses the questions to develop several questionnaires.

©2025 Pegasystems Inc

You create questions in bulk by editing a template Excel file, and then uploading the file to your application.

- **Populating a question template**
- **Importing a question template**
- **Troubleshooting question template errors**

- Adding a question to a question page
- Troubleshooting question template errors

Populating a question template

Define single-answer questions that you want to create in bulk by populating a question template. You download the template in your application, and add questions by following the structure of the sample questions that the template contains.

For example, define several questions about the work experience of a job candidate that you can then use to create a questionnaire in your application.

1. Download the question template:
   a. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.
   b. On the **Questions** tab, click **Import questions**.
   c. Click **Download template**, and then save the `QuestionsTemplate.xlsx` file to your local machine.
2. Open the `QuestionsTemplate.xlsx` file.
   The **Sample Questions** tab of the Excel file contains examples of correctly configured questions that you can import.
3. In the **ID** column, enter text that uniquely identifies the question.

   > **For example:**
   >
   > Enter `WorkHistory`.

©2025 Pegasystems Inc

If you enter the ID of an existing question, when you import the question template, your application prompts you to update or skip the question.

4. In the **Name** column, enter a label for your question.

> **For example:**
>
> Enter `Your work history`.

5. Provide text for your question:

Choices	Actions
Use plain text	a. In the **Question Layout** column, select **plain**. b. In the **Question Plain Text** column, enter the question text.
Use styled text	a. In the **Question Layout** column, select **rte**. b. 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.

6. Configure the answer format for your question:

Choices	Actions
A question that a user answers by attaching a file	a. In the **Answer Mode** column, select **File**.

©2025 Pegasystems Inc

Choices	Actions
A question that a user answers in a text area, text box, date, or date and time format	a. In the Answer Mode column, select freeform. b. In the Answer Freeform Format column, select an option that corresponds to your answer format.
A question that a user answers by selecting one or more available options	a. In the Answer Mode column, select checkbox.
A question that a user answers by selecting one of available options	a. In the Answer Mode column, select radio. b. In the Answer option labels column, enter a comma-separated list of the choices that users can select. **For example:** Enter `None,Basic experience,1-3 years on the job,Expert (more than 3 years on the job)`. c. In the Answer option values column, enter a comma-separated list of the values that map to the choices that users can select. **For example:** Enter `None,Basic,Advanced,Expert`.

©2025 Pegasystems Inc

Choices	Actions
	d. In the **Radio button label orientation** column, choose whether the your application displays choices stacked or side-by-side.
A question that a user answers by selecting one of available options in a drop-down list	a. In the **Answer Mode** column, select **dropdown**. b. In the **Answer option labels** column, enter a comma-separated list of the choices that users can select. **For example:** Enter `None,Basic,Advanced,Expert`. c. In the **Answer option values** column, enter a comma-separated list of the values that map to the choices that users can select. **For example:** Enter `None,Basic,Advanced,Expert`.
A question that a user answers in a custom section that you provide	a. In the **Answer Mode** column, select **section**. b. In the **Section Reference** column, enter the name of a section that is in the same class path as the question.

©2025 Pegasystems Inc

7. In the **Answer layout direction** column, choose whether your application displays the question and answer stacked or side-by-side.
8. To change where your application stores the answer to a question on the Pega Clipboard, in the **Answer Property** column, enter the name of a property that is in the same class path as the question.
9. To validate the answer that users provide at run time, in the **Validation** column, enter the name of a Validation Rule that is in the same class path as the question.
10. **Optional:** To add more questions to the template, repeat steps 3 through 9.
11. Save and close the file but do not change the file format.

> **What to do next:**
>
> Apply advanced functionality, such as conditional answers, that the question template does not support. For more information, see Fine-tuning your survey.

- Troubleshooting question template errors

Importing a question template

Automate the creation of single-answer questions by importing a question template. For example, the author of a marketing campaign provides a list of questions that you can add to a questionnaire in your application to collect information from the customers.

1. In the header of Dev Studio, click **Configure** > **Case Management** > **Questionnaire**.
2. On the **Questions** tab, click **Import questions**.
3. Click **Browse**.
4. Navigate to the location of your question template, and then upload the file by clicking **Open**.
5. **Optional:** To make the questions compatible with questionnaires that are outside your work pool, in the **Apply to class** field, provide the name of the class that contains your questionnaires.
6. Click **Next**.

©2025 Pegasystems Inc

> **Result:**
>
> The application validates the entries in your question template.

7. Click **Import**.
8. **Optional:** To review a summary of questions that your application created, updated, or skipped, click **View details**.
9. Click **Done**.

> **What to do next:**
>
> Add your questions to a question page in a questionnaire. For more information, see Adding a question to a question page.

- Troubleshooting question template errors
- Adding a question to a question page

Troubleshooting question template errors

When you use a question template to create questions in bulk, you might encounter errors. To resolve common issues that prevent you from importing multiple questions to your application, follow the respective guidelines for each scenario when the importing Process fails.

The following are the most common issues that can occur during bulk question imports:

- The template contains the IDs of questions that are already in your application
- The template has an invalid file structure or an invalid file type
- Questions in the template fail validation

Condition

The process of importing the question template fails and you receive an error message.

©2025 Pegasystems Inc

The template contains the IDs of questions that are already in your application

Solution: Solving duplicate questions

1. In the Duplicate questions section of the Import questions page, click View details.
2. Review the list of questions to confirm that you intentionally added them to the template.
3. Resolve the conflicts:
   - To exclude the duplicate questions from the import, select Skip.
   - To overwrite the existing questions with the information in the template, select Update.

   If you provide more than one entry with the same ID in the template, your application uses only the information from the first row that it finds.
4. In the **Validate and import** stage, click Import.
5. In the **Summary** stage, click Done.

The template has an invalid file structure or an invalid file type

Solution: Fixing the question template

1. On the **Import questions** page, click Download template, and then save the `QuestionsTemplate.xlsx` file to your local machine.
2. Open the `QuestionsTemplate.xlsx` file.
3. Copy the information from your original template to the relevant columns in the file.
4. Save the file but do not change its format.
5. In the **Upload file** stage, click Browse.
6. Navigate to the location of your new question template, and then upload the file by clicking Open.
7. Click Next.
8. In the **Validate and import** stage, click Import.

©2025 Pegasystems Inc

9. In the **Summary** stage, click Done.

Questions in the template fail validation

Solution: Fixing question configuration

1. In the Failed validation section of the Import survey questions page, click View details.
2. Determine which columns have invalid data or reference an invalid Rule by reviewing the error messages for each question.

   > **For example:**
   >
   > When the answer format for a question is set to Section but you do not provide the name of a Section Rule, you receive an error message.

3. Click Back.
4. Update your template to provide the required information.
5. Create the required Rules in the same class path as the questions.
6. Click Browse.
7. Navigate to the location of your updated question template, and then upload the file by clicking Open.
8. Click Next.
9. In the **Validate and import** stage, click Import.
10. In the **Summary** stage, click Done.

Related tasks

- Creating questions in bulk
- Populating a question template
- Importing a question template

©2025 Pegasystems Inc

Asking a question in a Case

Collect a single response from a user who processes a Case by using the **Question** shape in a Case Lifecycle. By isolating individual questions, you can selectively capture user input without the formal structure of a questionnaire.

For example, in a home insurance process, a Case worker asks a customer what kind of offers they want to receive in the future.

> **NOTE:** Only traditional UI Pega Platform applications support running a question from the Case Lifecycle outside of a questionnaire. In applications based on Constellation, you can run a questionnaire by using the **Questionnaire** shape. For more information, see Running a questionnaire in a Case.

> **Before you begin:**
>
> Create a question that you want to ask in the Case. For more information, see Adding a question to a question page.

1. Add the **Question** shape to the Lifecycle of your Case.
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. On the **Workflow** tab, in the **Lifecycle** section, hover over a Process in a Stage, and then click **Step** > **More** > **Automations** > **Question**.
   c. Click **Select**.
2. In the **Answer question** field, provide a name for the Step, for example, `Type of offers`.
3. Choose a question to ask:

©2025 Pegasystems Inc

Choices	Actions
Reuse an existing question	a. In the **Question type** list, select **Rule**. b. In the **Question rule** field, press the Down arrow key, and then select the name of a question. **NOTE:** You cannot select a complex question, because only questionnaires support complex questions. In the **Question Info** section, you can verify that the question is relevant to the Case. c. **Optional:** To ensure that the user provides an answer to the question, select the **Required** checkbox.
Create a virtual question **NOTE: Virtual questions are not reusable. Use this approach only when you are collecting a text response and the question is specific to the Process that contains the Question shape.**	a. In the **Question type** list, select **Text**. b. In the **Question text** field, enter the text of the question that your application displays at run time. c. In the **Map answer to property** field, press the Down arrow key, and then select a property to store the value of the answer that the user provides.

©2025 Pegasystems Inc

4. In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the question:
   - To route the question to the user who currently processes the Case, select **Current user**.
   - To route the question to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to answer the question.
5. **Optional:** To control whether your application displays the question at run time, in the **Display when** field, press the Down arrow key, and then select a When Condition.

   > **For example:**
   >
   > Display a question about a vehicle engine number only if the customer's insurance covers the type of accident that a customer had.

6. **Optional:** To validate the answer that the user provides, configure the processing to perform:
   a. In the **Pre-question processing activity** and **Post-question processing activity** fields, press the Down arrow key, and then select the name of an activity.

      > **For example:**
      >
      > Your application runs these activities when the user is online.

   b. In the **Pre-question javascript function** and **Post-question javascript function** fields, enter the names of functions that the harness of the Case includes.

      > **For example:**
      >
      > Your application runs these functions when the user is offline.

PEGA ©2025 Pegasystems Inc

> **NOTE:** Applications based on Constellation do not support pre-question and post-question processing. You can run these activities and JavaScript functions only in standard Pega Platform applications.

7. **Optional:** To provide additional comments about the question, in the **Audit note** field, enter some helpful extra text.
8. Click **Save**.

- Case Lifecycle elements
- Asking a group of related questions in a Case
- Running a questionnaire in a Case
- Designing questionnaires

Asking a group of related questions in a Case

Collect information from a user who processes a Case by using the **Question page** shape. By providing a structured format for questions and answers, you can quickly incorporate user feedback into a Case.

For example, in a car insurance Case, a Case worker asks a customer a set of questions about the circumstances of an accident, and then uses the answers to create a summary of the insurance claim.

> **NOTE:** Only traditional UI Pega Platform applications support running a **Question page** shape from the Case Lifecycle outside of a questionnaire. In applications based on Constellation, you can run a questionnaire by using the **Questionnaire** shape. For more information, see Running a questionnaire in a Case.

©2025 Pegasystems Inc

Before you begin:

Create a question page that you want to run in a Case. For more information, see Create a question page.

1. Add the **Question page** shape to the Lifecycle of your Case.
   a. In the navigation pane of Dev Studio, click **Cases**, and then click the Case Type that you want to open.
   b. On the **Workflow** tab, in the **Lifecycle** section, hover over a Process in a Stage, and then click **Step** > **More** > **Automations** > **Question page**.
   c. Click **Select**.
2. In the **Answer set of questions** field, provide a name for the Step, for example, `Rate the product`.
3. In the **Route to** field, press the Down arrow key, and then select the user who is responsible for answering the questions on the page:
   - To route the question page to the user who currently processes the Case, select **Current user**.
   - To route the question page to a specific user, select **Specific user**, and then, in the **Operator** field, enter the ID of the user to complete the question page.
4. **Optional:** To control whether your application displays the question page at run time, in the **Display when** field, press the Down arrow key, and then select a When Condition.

   For example:

   Display a set of questions about the details of a vehicle only if the customer's insurance covers the type of accident that a customer had.
5. **Optional:** To provide additional comments about the question page, in the **Audit note** field, enter some helpful extra text.
6. Click **Save**.

©2025 Pegasystems Inc

- Adding a question page to a questionnaire
- Running a questionnaire in a Case

Creating reports based on questionnaires in Dev Studio

Analyze and manage data from questionnaires in your Pega Platform applications by creating and configuring report definitions. For example, in a shopping application, you can view the responses from several questionnaires about customer hobbies at once, and display the data as a chart that helps you visualize the distribution of favorite pastimes in different age groups, to help you prepare a new sales offer.

> **NOTE:** In applications based on Constellation, you can visualize questionnaire data more conveniently in a low-code way by creating Insights on the **Explore Data** landing page. For more information, see Visualizing questionnaire data by creating Insights in Constellation.

> **Before you begin:**
>
> Create a questionnaire in your application. For more information, see Creating a questionnaire.

Because a questionnaire is a regular Case Type, you create and use report definitions for questionnaires in the same way as for any other Case Type.

1. In your application, create a report definition for your questionnaire Case Type.

   > **For example:**
   >
   > In a healthcare application, create a report definition with the label `Diabetes patients with high risk level` and columns that reflect the Data Model of your questionnaire Case Type. Then, you can define

©2025 Pegasystems Inc

> filters so that the report displays only patients who are diagnosed with diabetes, and whose risk level is considered high.

For more information, see Creating advanced reports.

2. **Optional:** To visualize data from the questionnaire and make analyzing the data easier, add a chart that presents the answers from your questionnaire.

> **For example:**
>
> Visualize hundreds of questionnaires from a study on high-risk patients as a column chart that groups these patients according to the quality of life that they reported.

For more information, see Adding charts in the Report Definition rule form.

- Creating a questionnaire
- Report definitions
- Adding charts in the Report Definition rule form
- Creating Insights

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.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

©2025 Pegasystems Inc

> For more information, see Creating a new application from a Blueprint.

- **Configuring Case Type access**
- **Controlling access to Case attachments**

Related information

- Managing concurrent Case access

Configuring Case Type access

The Access Manager **Work & Process** tab provides information about the authorizations to operations in the application that users have, based on Case Types 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 Process 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**
- **Editing authorizations for Case Type items in a single Access Group**

©2025 Pegasystems Inc

- **Viewing authorizations for Case Type items in all Access Groups**
- **Viewing authorizations for Processes and Flow Actions in a single Access Group**
- **Editing authorizations for Processes and Flow Actions in a single Access Group**

- Access Manager
- Managing concurrent Case access

Viewing authorizations for Case Type items in a single Access Group

From the Access Manager you can view settings for Access Groups other than your own. This allows you to review the authorization settings for Access Groups and Case Type items.

> **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, Process, or Flow Action in the Case Type.
- Operators in this Access Group who have no access to this Action, Process, or Flow Action in the Case Type.
- A combination of conditional access and possibly no access to this Action, Process, 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:

©2025 Pegasystems Inc

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.

©2025 Pegasystems Inc

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.

©2025 Pegasystems Inc

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

©2025 Pegasystems Inc

Viewing authorizations for Processes and Flow Actions in a single Access Group

Access Manager displays all Process and Local Flow Action defined for each Case Type in the selected application or applications. This view allows you to review the authorization settings for Processes 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 or Flow Action in the Case Type.
- Operators in this role have no access to this Process or Flow Action in the Case Type.
- Combination of conditional and possibly no access to this Process or Flow Action among operators in the Access Group.

To view the authorizations for Processes 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.

©2025 Pegasystems Inc

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
- Flow Actions

Editing authorizations for Processes and Flow Actions in a single Access Group

In Access Manager you can edit and change the authorization for Processes 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 Processes and Flow Actions, ensure 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.

©2025 Pegasystems Inc

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

Enforce security requirements and prevent users from inadvertently modifying Case data by controlling access to attachments in a Case. By applying security polices to Case content, you ensure that users only interact with attachments that are appropriate for their role in a business process.

Consider a scenario in which a loan applicant provides a set of documents that includes an ID scan, proof of income, and a financial history with previous loans. After you restrict user actions for attachments, only the customer service representatives who work on loan requests can view the content attached to the Case, instead of users with other roles in your application. Additionally, you can specify that only a manager can view the customer's financial history. To ensure that users always have access to all the documents required for a Case resolution, you can also prevent users from deleting any of the attachments.

You secure Case attachments by first categorizing attachments and then defining what actions users can perform on each category. For example, you can allow users to edit documents to ensure that the Case includes all relevant and up-to-date information.

PEGA ©2025 Pegasystems Inc

- **Categorizing Case attachments**
- **Attachment types**
- **Restricting user actions for Case attachments**
- **Extension points and supporting Rules for attachments**
- Performing a virus scan on an attachment
- Extension points and supporting Rules for attachments
- Setting required attachments for Stage entry

Categorizing Case attachments

To facilitate a more efficient resolution of your business goals, ensure that users have relevant data during Case processing by selecting attachments categories that a Case Type supports. When you categorize attachments, you define business classifications to provide access control policies that are specific to each attachment category. For example, first add a Classified category so that later you can define a group of users who can access the content of this category, such as managers.

You can save time by selecting an existing category or create new categories to address your business requirements. You can also specify whether an attachment of a specific category is required to resolve a Case. Consider a scenario for a loan application Case Type, in which a resolution is possible only when a Case contains a document with the proof of income of an applicant.

1. In the navigation pane of App Studio, click **Case Types**, and then click the Case Type that you want to open.
2. On the **Settings** tab, click **Attachment categories**.
3. Click **Add attachment category**.
4. In the **Category name** field, identify an attachment category.
    - To create a new attachment category, enter a unique name.
    - To use an existing attachment category, press the Down arrow key, and then select the name of the attachment category.

©2025 Pegasystems Inc

5. **Optional:** To ensure that a Case contains at least one attachment in the attachment category when a user submits the Case, select the **Required for resolution** checkbox.
6. Click **OK**.
7. Click **Save**.

Result:

The system applies your changes to all unresolved Cases. Users can select the attachment category when they add an attachment to a Case.

What to do next:

Define a set of actions that users can take on every attachment category. For more information, see Restricting user actions for Case attachments.

- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools

Attachment types

Attachment types define the media and format of items that users attach to Cases in an application.

The following standard attachment types are supported:

- **File** — Uploaded files of any type

  By default, a file attachment is saved as an instance of the *Data-WorkAttach-File* class, which maps to the `pc_data_workattach` database table. Optionally, application users can associate a category with each file attachment. The category describes the purpose or contents of the attachment.

- **Note** — Notes that contain only text

©2025 Pegasystems Inc

You can type or paste text into a note attachment. A note attachment is saved as an instance of the *Data-WorkAttach-Note* class. You cannot edit a note attachment after it is saved.

- **Screenshot** — Screen capture images

  Your application can include a window capture feature, so that users can capture a picture of any desktop window. Users can permanently save this image as an attachment to a Case (an instance of the *Data-WorkAttach-ScreenShot* class).

- **Scanned document** — TIFF files

  Users can scan documents by using Pega Document Scanning or the Image Viewer feature and attach the TIFF files to a Case. A scanned document attachment is saved as an instance of the *Data-WorkAttach-ScanDocument* class. The standard *Work-.pyCreateScannedDocument* local Flow Action starts this operation. The user can also configure scanner defaults from this Flow Action.

- **URL** — A single URL

  A URL attachment links the Case to a web page or other object that is accessible with the HTTP protocol. A URL attachment is saved as an instance of the *Data-WorkAttach-URL* class.

- **Content** — Content that is saved in an external Enterprise Content Management (ECM) system

  You can configure your application to store attachments of type **file** in an external ECM system by using the CMIS protocol. A content attachment is saved as an instance of the *Data-WorkAttach-ECM* class.

- Performing a virus scan on an attachment
- Restricting user actions for Case attachments
- Attaching content to a Case
- Limitations for attachments in a file storage repository

PEGA ©2025 Pegasystems Inc

Restricting user actions for Case attachments

Improve Case security by ensuring that users can interact with content that is appropriate for their role only. When you define Conditions or privileges in an attachment category, you control which actions a user can take on a Case attachment. For example, when working with a loan request Case Type, only a manager can access documents that contain customer's sensitive data.

Before you begin:

Define attachment categories that your Case Type supports. For more information, see Categorizing Case attachments.

NOTE: At run time, restricting actions works for attachments of file and URL types only.

1. In the navigation pane of Dev Studio, click **App**.
2. In the class of the Case Type that you want to configure, expand the **Process** section.
3. Click **Attachment Category**, and then click the name of the attachment category that you want to configure.
4. Identify the Case attachments types that your attachment category supports:
   a. Click the **Availability** tab.
   b. Select the checkbox next to one or more relevant attachment types.
5. On the **Security** tab, restrict user actions on the attachment types:

Choices	Actions
Restrict user actions accordingly to a privilege	a. In the **Access control list by privilege** section, in the Privilege field, enter a privilege that you want to use to grant user actions.

©2025 Pegasystems Inc

Choices	Actions
	b. Select a checkbox in one or more columns, based on the user operations that this privilege grants. **For example:** To allow users to delete attachments that they create, select the **Delete own** checkbox. c. To add more privileges and specify other actions for each privilege, click **Add privilege**, and then repeat steps 5.a and 5.b. **Result:** At run time, users with the specified privileges can perform actions that you assign to each privilege.
Restrict user actions accordingly to a When Condition	a. In the **Access control list by When Rule** section, in the **When** field, enter a When Condition that you want to use. **For example:** Select a When Condition that at run time evaluates if a

©2025 Pegasystems Inc

Choices	Actions
	user belongs to a `Managers` Access Group. b. Select a checkbox in one or more columns, based on the user operations that this privilege grants. **For example:** To allow users to add attachments to a Case, select the **Create** checkbox. c. To add more When Conditions, click **Add when**, and then repeat steps 5.a and 5.b. **Result:** At run time, if a When Condition evaluates to true, users can perform actions that you associate with the When Condition.

NOTE: The absence of a privilege or When Condition does not automatically restrict a user operation. For example, if you define a Condition that allows users to create attachments, ensure that you also define another Condition that restricts editing, viewing, and deleting attachments.

©2025 Pegasystems Inc

6. **Optional:** To allow users to choose which teams can access the attachments that the users provide, select the **Enable attachment-level security** checkbox.
7. Click **Save**.

Result:

A user can perform an action only when all the When Conditions return a true value, the user belongs to the required team, and the user holds at least one of the required privileges.

What to do next:

Grant a privilege to Case workers who resolve Cases that include create permissions for this attachment category. For more information, see Granting privileges to an Access Role.

Related tasks

- Performing a virus scan on an attachment
- Setting required attachments for Stage entry
- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools
- Creating a Rule specialized by circumstance

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:

©2025 Pegasystems Inc

Name	Description
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.
GetAttachmentReference	An activity that you can override to customize attachment download.
Initialization/MaximumFileUploadSizeMB	An attribute that you can set in the *MaximumFileUploadSizeMB* dynamic system setting that you create to change the default size limit of 1 GB for attachments. For more information on how to create a dynamic system setting, see Creating a dynamic system setting. The following figure shows a sample customization of the *MaximumFileUploadSizeMB* dynamic system setting: *Dynamic system setting configuration for maximum attachment size* After you create the dynamic system setting, the **Value** field helps you set the maximum attachment size. The following figure shows

©2025 Pegasystems Inc

Name	Description
	a dynamic system setting that allows a maximum attachment size of 512 MB: *Setting the value for the maximum attachment size*
pc_data_workattach	The default table in the database that you can query to retrieve Case attachments. You can also store Case attachments in external Content Management Interoperability Services (CMIS) systems or web storage providers.
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.
pyMaxDragDropAttachSizeMB	An application-level dynamic system setting that defines the maximum file size (in megabytes) allowed for drag-and-drop attachments. If a user attempts to upload or download a file that exceeds this limit, the operation is blocked. The default value for this setting is 30 MB, which prevents attaching or retrieving files larger than 30 MB through drag-and-drop. Adjust this setting carefully if your application integrates with large file repositories and needs to support attachments larger than 30 MB.
pzMultiFilePath	A control that you use to upload multiple file attachments at once. The maximum size of

©2025 Pegasystems Inc

Name	Description
	attachments that you can upload with the control is defined in the *pyMaxDragDropAttachSizeMB* application setting. When you add the *pzMultiFilePath* control to a Section, you must also add the *pzAttachFileDDFileList* out-of-the-box Section to the same Section. The auxiliary *pzAttachFileDDFileList* Section displays the file attachment list and a picker for defining the file category. The following figure shows a sample scenario with the *pzMultiFilePath* control and the *pzAttachFileDDFileList* Section: *Run-time view of the pzMultiFilePath control and the pzAttachFileDDFileList Section* For more information, see Creating an embedded Section and Sections.
Work-.EditAttachment	A Flow Action that you can add to a Process so that users can open and edit file

©2025 Pegasystems Inc

Name	Description
	attachments in a Case. When you submit the form for this Flow Action, your application creates a new version of the attachment.

- Performing a virus scan on an attachment
- Attaching content to a Case

Best practices for compliance and updatability

By following the best practices for compliance and updatability, you create efficient, sustainable, and guardrail-compliant applications that help you deliver business outcomes faster and more consistently.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.
>
> For more information, see Creating a new application from a Blueprint.

You can apply the following best practices to ensure that your Case Types support reuse and function as expected with subsequent Pega Platform™ updates:

- Add every Case Type to the list on the **Cases & data** tab of the application record, with the appropriate configuration to support its creation.

  For more information, see Managing application Cases and data.

- Ensure that every Case Type has Stages, even if the Stages are only `Create`, `Open`, and `Resolved` that are analogous to the create, read, update, and delete (CRUD) operations.

©2025 Pegasystems Inc

For more information, see Stages in a Case Lifecycle.

- Ensure that every Case Type has a `Create` Stage and at least one `Resolution` Stage.

  For more information, see Stages in a Case Lifecycle.

- Avoid using temporary Cases. Instead, use the `Create` Stage.

  For more information, see The Create Stage.

- Define entry conditions for Stages and gather related details for each Stage before the Case enters the Stage.

  For more information, see Defining the entrance criteria for a Stage.

- Use the option to update the Case Status during the Case Lifecycle.

  Case Statuses help communicate Case progress to the users. For more information, see Changing Case Statuses.

- Use the standard prefixes `New-`, `Open-`, `Pending-`, and `Resolved-` for all statuses.

  For more information, see Case Status values.

- Clearly define the complete Microjourney® Lifecycle to the correct level of granularity.

  If the Lifecycle becomes large or complex, consider wrapping logical items into a Subprocess or Subcase and instantiating from the top-level Lifecycle. For more information, see Creating a Microjourney for customer success.

- In applications that you build with Theme Cosmos, consider relying on Subprocesses while processing a list of Embedded Data objects.

PEGA ©2025 Pegasystems Inc

This approach facilitates running common logic against each list item. For more information, see Embedding data in a Case.

- List all participants that are directly involved in the Case as parties.

  For more information, see Defining Case participants.

- Include Personas, Channels, as well as system and data methodology where required.

  For more information, see Understanding Personas and access management.

- Avoid hardcoding Actions directly into the UI; instead, use standard Case processing or Optional Case-wide and Stage-wide Actions in the Case Lifecycle.

  For more information, see Configuring and working with optional Actions in Cases.

- Avoid using tickets; instead, use features such as the Wait shape.

  For more information, see Pausing and resuming Processes in Cases.

- Use standard routing features.

  For more information, see Assigning tasks to users.

- Identify the list of Case Types whose instances the system associates with one another at run time.

  A Case and its association with other Cases provide a more comprehensive understanding of how various Cases are related. For more information, see Understanding Case hierarchy.

- Create child Cases when your work can be broken down into multiple pieces that have a relationship or dependency with one another. Use the following criteria to identify when to use a child Case:
  - There is a distinct Case Lifecycle from the top-level Case Type.

PEGA ©2025 Pegasystems Inc

- This work can be run independently or by many top-level Case Types.
- You need to report on the work independently.
- This work will progress in parallel to other child Cases or the top-level Case.

For example, a Case Type that represents opening a new bank account requires that the applicant has a credit check. Opening a new bank account is visualized in a Case Type, but the credit check has its own Case Life Cycle that is critical to the overall Microjourney® of opening an account. The Credit Check should be a child Case because a credit check can run independently from other Processes in opening a bank account. If you separate the credit check into its own child Case, it can be reused by other top-level Case Types, such as applying for a loan. For more information, see Creating a child Case Type in Case Designer and Understanding Case hierarchy.

- Use standard Views and design templates, and ensure that all Assignment forms are configurable with the Configure form.

  For more information, see Designing the UI in Case Types.

- Add correct validation to forms that require it, and add Lifecycle validation on the **Data Model** tab of a Case Type.

  Form validations ensure data integrity regardless of the Channel through which the submission of data occurs. For more information, see Restricting user input in a field and Configuring Case-wide validation for correct data entry in fields.

- Use attachment fields when you expect users to attach and view a file or document as part of the tasks that are assigned to the users.

  For more information, see Attaching content to a Case.

- Define custom attachment categories and avoid relying only on the out-of-the-box categories.

PEGA ©2025 Pegasystems Inc

Explicit categorization helps you group related files and use the reference while sending out attachments by email. For more information, see Categorizing Case attachments.

- Consider defining relevant Service-Level Agreements (SLAs) for every task to ensure that users get the appropriate amount of time to work on a task.

  You can also define SLAs for a Process, Stage, or Case. For more information, see Completing work on time.

- Consider relying on Urgency to convey the importance of a task.

  Higher values indicate higher importance. You can manage Urgency through a Local Action or an SLA. For more information, see Urgency.

- Use only the fields that enable tracking and auditing changes (field-level audit).

  For more information, see Enabling field-level auditing.

- When you need multiple users to access a Case simultaneously, consider setting the locking strategy to allow multiple users.

  For more information, see Managing concurrent access to a Case.

- Provide a few default tags that users can use while working on a Case.

  Relying on the tag infrastructure provides an easier way for associating multiple Cases. For more information, see Tagging in Cases.

- Configure additional feed sources in addition to standard Pulse notifications so that users have a comprehensive overview of information related to a Case.

  For more information, see Adding feed sources to activity feeds and Configuring Pulse feed sources for Constellation.

PEGA ©2025 Pegasystems Inc

- Use the Send Notification shape and Notification framework to send Notifications, especially when targeting large groups.

  For more information, see Sending event notifications from Cases.

- Extend standard Pulse Notifications to send updates to specific users who are related to the Case.

  By default, Pega Platform sends Notifications to followers, @ mentions, and users who participate in a thread. For more information, see Configuring Pulse email notifications.

- Mark important properties, Actions, Rules, and other assets as relevant records.

  As a result, you improve the authoring experience in App Studio by curating records that you need to include in your Case Types and data types. For more information, see Relevant records for Rule reuse.

Related concepts

- Understanding and complying with Case Management best practices

Fixing Cases in flight with Problem Flows

In Pega Platform™, when a user works on a Case and an error arises, the system automatically initiates a Problem Flow, which is a special error-handling Process that provides an interface to help facilitate the detection of problems, and analysis or repair of such events.

> **NOTE:** As a best practice, use Pega GenAI Blueprint™ to suggest optimal Case Type designs instantly, based on GenAI and a vast repository of workflow best practices.

©2025 Pegasystems Inc

For more information, see Creating a new application from a Blueprint.

Issues that result in Problem Flows

Issues can arise due to a multitude of reasons, ranging from missing Rule dependencies to data integrity issues. Typically, the following Process and Stage configuration issues and integration errors might start a Problem Flow:

- Shape configuration problems, typically with an Activity or Automation referenced in a shape.
- The Skip Stage, when the Stage entry validate mechanisms fail.
- Decision failure in the Wait shape, when none of the connectors between shapes can be chosen. The run of the Process becomes stuck after a Decision shape, or none of the outgoing Connectors evaluate to true.
- Assignment routing failure, when an Assignment becomes corrupted and does not match up with the Case to which it points.
- A Connector Rule fails or times out.
- Correspondence failure, when correspondence is not being sent, for example, due to a bad email address.

Standard Problem Flows

Pega Platform provides the following standardized Problem Flows for the most common types of issues:

- *FlowProblems* for general Process-related configuration issues.
- *ConnectionProblem* for integration and Connector-related issues.
- *pzStageProblems* for Stage-related configuration issues.
- *pyRemoteProblem* for problems with creating Cases in a remote system.

NOTE: To support your specific requirements, you can customize the default Problem Flows.

©2025 Pegasystems Inc

> For more information, see the Handling flow changes for Cases in flight module on Pega Academy.

Problem Flows: User Journey

Running into a workflow problem:

When a run-time exception occurs, the system automatically starts a Problem Flow. The Case worker can view the link to the Problem Flow Assignment, but only an administrator or anyone in their Access Group who has access to the *BrokenProcesses* Work Queue can open and process the Problem Flow Assignment.

Workflow error starting a Problem Flow in a Case

Fixing the workflow problem:

An administrator clicks **Go** to open the Problem Flow related to the Case in flight, in which they can view error details, such as the Stage, Process, or Shape in which the exception occurred, as well as the error message.

Having the necessary information, the administrator can then decide whether the problem is something that they can resolve. If they can, the administrator makes the necessary changes by stepping out of the Problem Flow into Dev Studio and fixing the flawed Rule or editing the Clipboard.

©2025 Pegasystems Inc

For more information, see Troubleshooting shape-related configuration issues for Cases in flight.

If the administrator cannot deal with the exception, they contact the Support Center. For more information, see Pega Support Center.

Restoring Case processing after a fix is implemented:

After resolving the root cause of the issue, the administrator goes back to the Case with the open Problem Flow and performs one of the following standard actions:

NOTE:

- The recommended action for restoring Case processing is available in the work area of the current Case, but you can also use other dedicated actions that are available in the **Actions** menu of the Case:

Flow actions for restoring Case processing in the Case UI

- The availability of particular actions depends on the type of Problem Flow that the system runs to solve a flow error. For example the **Restart Problem Flow from the beginning** action is always available, while the **Resume (fixed) problem flow** action is available for the Problem Flows related to general Process-related issues but not for connector-related issues.

©2025 Pegasystems Inc

Restart problem flow from the beginning

This action resumes the Case at the initial Step, when the update was made to a Step that the application has already processed.

Retry last Step

This action resumes the run of the flow by running the Step that paused.

This action is usually run after fixing a connector timeout or failure, or if you resolve an error by reconfiguring a Rule used by the flow. For example, if you add a missing result to a Decision table to fix a flow error, select the **Retry last Step** action to re-evaluate the Decision table and determine the appropriate flow connector.

Resume (fixed) problem flow

This action resumes the run of the flow that begins at the Step after the paused Step.

This action is usually run after fixing a shape issue. For example, a flow contains a Decision shape that evaluates a Decision table. If the Decision table returns a result that does not correspond to a connector in the flow, add the connector to the Decision shape and then resume it. The flow uses the already-determined Decision result to select the appropriate connector and advance to the appropriate shape.

Cancel this Assignment

This action removes the Assignment.

> **NOTE:** You can customize the Problem Flows to include an activity that sends an email notification to managers or other stakeholders when an Assignment is canceled. For more information, see the Resolving flow issues by using problem flows topic on Pega Academy.

©2025 Pegasystems Inc

Restart Process

This action helps remediate exceptions where the **Retry the last step** action is not a suitable remediation action.

For example, a Process that references a Rule that does not exist requires the **Restart Process** action for remediation. The action reruns the paused Case from the start of the current Process.

> **NOTE:** If the exception occurs in a stand-alone Process, the action restarts the stand-alone Process. If the exception occurs in a Subprocess, the action restarts the Subprocess. For more information, see Subprocesses.

When the admin task is done, they can route the Assignment back to the Case worker with the **Transfer assignment** optional user action in the **Actions** menu of the Case, so that the Case worker can resume Case processing. For more information, see Transferring Assignments.

> **NOTE:** Administrators can also manage Problem Flow Cases from a dedicated landing page. For more information, see the Managing problem flows on the Flow errors page topic on Pega Academy.

- **Troubleshooting shape-related configuration issues for Cases in flight**

Related concepts

- Understanding Personas and access management
- Troubleshooting final Rules
- Troubleshooting REST connectors
- Troubleshooting your system

PEGA ©2025 Pegasystems Inc

Related reference

- Automation troubleshooting
- Troubleshooting newly created Rules

Troubleshooting shape-related configuration issues for Cases in flight

Fix Cases that stall when a shape run fails to restore workflow efficiency and reduce operational delays.

Condition

When an error caused by a shape-related configuration issue occurs during Case processing, the system initiates a standardized *FlowProblems* Process to assist in problem detection and resolution. The error-handling Process manifests itself in the stalled Case UI as the Problem Flow Assignment but is also accessible through a dedicated landing page for managing Problem Flows in your application.

Cause

The following list describes sample conditions that can result in starting the Problem Flow because of a shape configuration issue:

- A shape references a flawed Activity or Automation.
- A shape references a non-existent Rule, or a Rule that the Case worker or background processing is running in a shape lacks the required Rulesets, versions, Access Roles, Privileges, or other prerequisites for implementation.
- A shape references a non-existent data instance, for example, a Work Queue, Operator, Organization, Work Group, or Access Group.

In this sample scenario, the administrator encounters and fixes a problem with a Case by using a Utility shape. It turns out that the shape configuration contains a flawed Activity, which, in this scenario, divides the value of the referenced property by zero.

PEGA ©2025 Pegasystems Inc

Solution

Responsible party

The administrator or anyone in their Access Group who has access to the *BrokenProcesses* Work Queue. For more information, see Understanding Personas and access management and Learning about Access Groups.

NOTE:

- The administrators can begin fixing the stalled Case in one of the following areas:
  - The stalled Case UI by accessing the Problem Flow Assignment, as shown in the following figure:

Problem Flow Assignment in a stalled Case

    For more information, see Fixing Cases in flight with Problem Flows.
  - A dedicated landing page for managing Problem Flows.

    For more information, see the Managing problem flows on the Flow errors page topic on Pega Academy.

- Depending on the place where the fix begins, the names of the actions for restoring Case processing after the fix might slightly differ, but represent the same operations to restore Case processing. For example, in the Case UI, one such action name is *Resume (fixed) Problem Flow*, while on the landing page it is *Resume Flow*.

©2025 Pegasystems Inc

This sample scenario shows a Problem Flow management from the dedicated landing page.

1. In the header of Dev Studio, click **Configure** > >**Case Management** > **Processes** > >**Flow Errors**.
2. On the **Flow Errors** tab, in the row with the Problem Flow that is related you your Case ID, check the error message, Process, and the Step in which the error occurred.

For example:

The following figure shows the Problem Flow with the error logs for the sample Case:

Problem Flow with the flawed Process, Step, and error message indication

3. In the **Flow** column, click the broken Process ID, fix the flawed element in the Process, and then save the changes.

For example:

Click **Review_Flow**, and then in the Process Modeler, double-click the Utility shape. Fix the Activity configuration in the shape by editing the flawed Rule:

©2025 Pegasystems Inc

Fixing a shape issue in the original Process

4. On the Problem Flows landing page, ensure that the appropriate Problem Flow is selected, and then in the table header, click one of the Flow Actions.
   For more information about the available Flow Actions, see Managing problem flows on the Flow errors page.

For example:

Click **Resume Flow**.

Result:

The Flow Action completes with a completion status message, as shown in the following figure:

©2025 Pegasystems Inc

> *Completion status message for the flow action*
>
> If resuming the Problem Flow resolves the issue in the original Process, the stalled Case process returns to just after the last successfully completed Step, runs a connector from that point, and then resumes processing for the Case worker.

5. If the Case worker has no admin privileges, route the Assignment back to the Case worker by clicking **Transfer assignment** > **Actions**.
   For more information, see Transferring Assignments.

Related concepts

- Configuring complex Processes
- Preparing for modifying Rules

Related tasks

- Processes in a Case Lifecycle
- Adding an automated Step to a Process

Related reference

- Automation troubleshooting

Interacting with Cases at run time

Attain your business objectives by creating and managing Cases in your application. Learn how to use tools that enable collaborative work on Cases.

- **Creating and managing Cases**
- **Collaborating on Cases**

©2025 Pegasystems Inc

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 customer service representative 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 Cases 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.

Creating Cases

You can configure Case creation option in the following ways:

In the Create Case Step in the Case Lifecycle

The system automatically starts a new Case when the current Case reaches the **Create Case** Step.

For more information, see Starting child and Remote Cases from the main Case context.

In the navigation pane of a Portal

Users click **Create** in the navigation pane of a Portal, and then select the type of the Case to create.

For more information, see Creating Cases from the navigation pane of a Portal.

In the Actions menu of the current Case

Users select a relevant Action that starts a Case from the **Actions** menu of the current Case.

For more information, see Creating Cases from the Actions menu of another Case.

PEGA ©2025 Pegasystems Inc

In a table in a Case or landing page

Users create Cases directly from a Case or landing page with a table.

For more information, see Creating Cases from tables.

In the bulk processing option in a Portal

In traditional UI applications, users create Cases in bulk by clicking the user icon, and then selecting **Bulk Actions** > **Create**. For more information, see Creating Cases in bulk.

In Constellation applications, users create Cases in bulk by clicking the button for bulk processing in a table header. For more information, see Enabling bulk processing of Cases in Constellation.

Managing Cases

Pega Platform™ offers a wide range of tools to help users manage and resolve work in a convenient and effective way. The availability of the tools varies depending on the Portal that users can access. When you create a new application, Pega Platform automatically generates the first Portal for you. Depending on your UI architecture, the default Portals are Web Portal for Constellation applications or User Portal for traditional UI applications. For more information about the types of applications that you can build in Pega Platform, see Application UI setup.For more information about the key components of the Constellation UI, see Constellation taxonomy and Constellation UI. For more information about the key components of the traditional UI, see Theme Cosmos taxonomy and Traditional UI.

- **Bulk processing Cases**
- **Creating ad hoc Assignments**
- **Creating ad hoc Cases**
- **Creating Cases from tables**
- **Creating Cases from the navigation pane of a Portal**

PEGA ©2025 Pegasystems Inc

- **Creating Cases from the Actions menu of another Case**
- **Following Cases**
- **Interacting with parent and child Cases on multiple tabs**
- **Out-of-the-box Portal features for creating and managing Cases**
- **Reopening resolved Cases**
- **Setting notification preferences**
- **Tagging in Cases**
- **Transferring Assignments**
- **Insights in Case Management**

Related reference

- Out-of-the-box Portal features for creating and managing Cases

Bulk processing Cases

Process Cases in bulk to save time and resolve work faster. Bulk processing is less error-prone than individual Case processing.

For example, managers can use bulk processing to transfer a large number of Cases to a specific user for processing, update the Urgency for multiple Cases, or send out a message about many Cases to a specific user.

Developers can instantly apply common Flow Actions to many Cases. For example, they can adjust Service-Level Agreement times and add a note to all Cases at once.

PEGA ©2025 Pegasystems Inc

> **NOTE:** Constellation applications support bulk Actions only for tables and lists that appear on landing pages and Case Pages. For more information, see Configuring a list-based landing page and Configuring list Views.

Related concepts

- Understanding Case hierarchy
- Queue processor Rules
- Pega-ProCom agents

Related tasks

- Configuring Case-wide validation for correct data entry in fields

Creating Cases in bulk

Create Cases in bulk to define Case details, such as labels and work statuses, in one step without manually interacting with individual forms. For example, in an insurance application, you can save time by creating multiple Cases for home, car, and life insurance in bulk.

> **NOTE:** Constellation applications support bulk Actions only for tables and lists that appear on landing pages and Case Pages. For more information, see Configuring a list-based landing page and Configuring list Views.

> **Before you begin:**
>
> - For traditional UI applications, ensure that your Access Group is associated with a role that grants you the *Work-.PerformBulk* privilege. For more information, see Granting privileges to an Access Role.

©2025 Pegasystems Inc

- For Constellation applications, ensure the set up is ready. For more information, see Enabling bulk processing of Cases in Constellation.

1. In an end-user Portal, in the rightmost end of the navigation pane, click the user icon (for example), and then select **Bulk Actions**.
2. In the **Bulk** section, click **Create**.
3. In the **Case Type** list, select the name of a Case Type.

> **NOTE:** Pega Platform supports bulk creation only for Parent Case Types.

4. Click **Add Case**, and then define values for your Cases:
   a. In the **Label** field, enter the name of the Case that is visible in Worklists and forms.

   > **Result:**
   >
   > The application stores labels in the *Work-.pyLabel* property.

   b. In the **Description** field, enter the purpose of the Case.

   > **Result:**
   >
   > The application stores descriptions in the *Work-.pyDescription* property.

   c. In the **Status** list, select the initial status of the Case.

   > **Result:**
   >
   > The application stores statuses in the *Work-.pyStatusWork* property and updates them automatically when Case workers process a Case.

5. **Optional:** To create more Cases, repeat step 4.

©2025 Pegasystems Inc

6. Click **Create Cases**.

Result:

The application creates Cases based on the values that you provide, and displays the current status of each Case in the **Processing Results** column of the Case list.

Case type *
Insurance offer

ID	Label	Description	Status	Processing Results
I-188001	Car insurance PREMIUM	UPlusTelco car insurance premium offer for a private customer	Open	Succeeded
I-188002	Car insurance regular	UPlusTelco car insurance regular offer for a private customer	New	Succeeded
I-188003	Home insurance PREMIUM	UPlusTelco home insurance premium offer for a private customer	New	Succeeded
I-188004	Home insurance regular	UPlusTelco home insurance regular offer for a private customer	New	Succeeded
I-188005	Life insurance PREMIUM	UPlusTelco life insurance premium offer for a private customer	New	Succeeded
I-188006	Life insurance regular	UPlusTelco life insurance regular offer for a private customer	Open	Succeeded

+ Add Case

Create Cases Clear

Cases that are created in bulk

Updating Cases in bulk

Update Cases in bulk to set the values of Case labels and descriptions in one step, without manually interacting with individual forms. As a result, you save time and reduce the risk of small errors in Case details.

Consider an example of an insurance sales application. After customer service representatives (CSRs) make hundreds of insurance offers to existing customers in a month, as part of a new marketing campaign, the company updates all offers from the previous week to the premium level. CSRs can update the labels and descriptions of all the eligible Cases in bulk, and save time by avoiding manual changes in each Case separately.

PEGA ©2025 Pegasystems Inc

NOTE: Constellation applications support bulk Actions only for tables and lists that appear on landing pages and Case Pages. For more information, see Configuring a list-based landing page and Configuring list Views.

Before you begin:

- To define more Case details during a bulk update, customize the *pyBulkUpdate* Section by adding relevant properties.
- To be able to reopen or update resolved Cases and ignore the validation criteria that are defined in a Case Type, bypass Validation logic by adding a role to your Access Group that grants you the *pxIgnoreBulkValidations* privilege. For more information, see Granting privileges to an Access Role.

1. In an end-user Portal, in the rightmost end of the navigation pane, click the user icon (for example), and then select **Bulk Actions**.
2. In the **Bulk** section, click **Update**.
3. In the **Case Type** list, display relevant Cases by selecting the name of a Case Type.
4. **Optional:** To refine the list of Cases, add filter criteria:
   a. Click **Add Filter**.
   b. In the **Filter Property** field, enter the name of the property that you want to evaluate in each Case.

      For example:

      To evaluate the Urgency of Cases, enter `Urgency`.

   c. In the **Condition** list, select a comparator.
   d. In the **Filter Value** field, enter a case-sensitive value to compare with the value of the **Filter Property** field.

©2025 Pegasystems Inc

For example:

Enter `80` to show Cases with the Urgency larger than 80, as shown in the following figure:

Entering filter criteria

e. **Optional:** To add more filter criteria, repeat steps 4.a through 4.d.

f. Click **Filter Cases**.

Result:

The application displays Cases that meet all the filter criteria.

5. Update values in the **Name** and **Description** columns.

For example:

To ensure that Case workers who process existing Cases apply new regulations, in the **Name** column, enter `Add pics - new customer`, and then, in the **Description** column, enter `Adding pictures for new customers is now subject to additional privacy regulations.`

6. Click **Update**.

Result:

The application updates Cases based on the values that you provide, and displays the current status of each Case in the **Processing Results** column of the Case list.

©2025 Pegasystems Inc

Running Flow Actions in bulk

Run the same Flow Action for multiple Cases at once by using the bulk Actions form. You can work on many Cases without impacting their path toward resolution. For example, you can extend a Service-Level Agreement (SLA) or transfer a set of Assignments to a user.

> **NOTE:** Constellation applications support bulk Actions only for tables and lists that appear on landing pages and Case Pages. For more information, see Configuring a list-based landing page and Configuring list Views.

1. In an end-user Portal, in the rightmost end of the navigation pane, click the user icon (for example), and then select **Bulk Actions**.
2. In the **Bulk** section, click **Process**.
3. In the **Process Cases** section, modify the filter criteria to refine the list of Cases: The default filter criteria return a list of Cases with at least one open Assignment that is assigned to you.
   a. Click **Add Filter**.
   b. In the **Filter Property** field, enter the name of the property that you want to evaluate in each Case.
   c. In the **Condition** list, select a comparator.
   d. In the **Filter Value** field, enter a Case-sensitive value to compare with the value of the **Filter Property** field.
   e. **Optional:** To add more filters, repeat steps 3.a through d.
   f. Narrow the list of results by clicking **Filter Cases**.

   > **For example:**
   >
   > You can find Cases that were recently created by comparing the **Create Date/Time** property with a specific date, as shown in the following figure:

©2025 Pegasystems Inc

Entering filter criteria

4. Select the Cases for which you want to run the Flow Action:
   - To select individual Cases, select the checkboxes next to the Cases for which you want to run the Flow Action.
   - To select all Cases, select the **Select all displayed results** checkbox in the top row of the list of results.

> **NOTE:** If your filter criteria return more than one page of results, to select all of the results, you need to select the **Select all displayed results** checkbox on each page.

Selecting all results on multiple pages

5. Click **Select Action**, and then select the Flow Action to run.
   The list of available Flow Actions depends on the Flow Actions that apply to the Cases that meet your filter criteria.
6. In the Flow Action dialog box, enter values in the relevant fields.

©2025 Pegasystems Inc

For example:

In the **Transfer Assignment** dialog box, to reassign a Case to a team, in the **Transfer to** list, select **Team**, and then provide the team name. In the **Comments** field, explain why you want the team to complete the Assignment.

7. **Optional:** To continue working while your application processes the Cases, select the **Bulk process in background** checkbox.
   Some Flow Actions support background processing, which can help you complete your work faster when you run a Flow Action for many Cases. As a result, an agent queues each request and sends you an email when the processing finishes.
8. Run the Flow Action by clicking **OK**.

Result:

The Flow Action runs for each Case, based on the values that you provide. You can view the status of the Action in the **Processing Results** column of the Case list.

NOTE: After the Flow Actions complete, your application does not clear the checkboxes next to the Cases that you selected. As a result, you can quickly run another Flow Action for the same Cases.

Related tasks

- Configuring indicator for a Flow Action

©2025 Pegasystems Inc

Troubleshooting errors in bulk processing

Resolve common issues that prevent you from processing multiple Cases by following the respective guidelines for each problem scenario.

Condition

Users want to create or update Cases, or run Flow Actions on multiple Cases. One of several problems prevents users from processing Cases.

Cause: The Operator menu of your Portal does not contain the Bulk actions option.

> **NOTE:** Constellation applications support bulk Actions only for tables and lists that appear on landing pages and Case Pages. For more information, see Configuring a list-based landing page and Configuring list Views.

Solution: Granting the appropriate privileges to your operator.

1. Associate a role with your Access Group that grants you the *Work-.PerformBulk* privilege. For more information, see Granting privileges to an Access Role.
2. Refresh your browser session.

Cause: The list of items to bulk process is read-only because another user has a lock on some Cases.

Solution: Coordinating your work on Cases with other users.

1. Hover over the **Lock** icon in the list of Cases to see the name of the user who currently has the Case open.
2. Contact the user to coordinate an appropriate time for you to bulk process the Case.
3. After the user releases the lock, click **Update** to refresh the list.

PEGA ©2025 Pegasystems Inc

Cause: The Processing Results column displays a failed status.

Solution: Resolving bulk processing errors.

1. Hover over the failed status to view an error message.
2. Follow the instructions in the error message to resolve the issue.
3. Rerun the bulk action.

Cause: The Processing Results column never displays a status.

Solution: Enabling the bulk processing agent.

1. In the navigation pane of Dev Studio, click Records.
2. Expand the SysAdmin category, and then click Agents.
3. In the list of agents on your node, click the *Pega-ProCom* agent.
4. On the Schedule tab, in the Scheduled Agents list, add the *AgentBulkProcessing* activity, and then select the Enabled checkbox for this activity.
5. In the **Agent-Wide Settings** section, select the Enable this agent checkbox.
6. Click Save.

Cause: The Instruction column displays errors.

Solution: Resolving Assignment errors.

1. In the header of Dev Studio, click Configure > Case Management > Tools > Work Admin > Worklist Assignment Errors.
2. In the **Assignments with errors** window, review the list of Problem Assignments and their corresponding error messages to determine which errors you can clear.

Cause: You receive no notifications when background processing is complete.

Some Flow Actions support background processing, which can help you complete your work faster when you run a Flow Action for many Cases. As a result, an agent queues

©2025 Pegasystems Inc

each request and sends you an email when processing finishes. For more information, see Queue processor Rules.

Solution: Configuring the email account

1. Ensure that the *Default* email account is correctly configured in your application. For more information, see Configuring outbound email.
2. On your Operator form, verify that the **Email** field contains a valid email address.

Cause: The list of Flow Actions in the Select Action list is incomplete.

> **NOTE:** The **Select Action** list displays only Flow Actions that are visible to your current work pool and that have a common ancestor class to the Cases that you are processing in bulk.

Solution: Configuring the display of Flow Actions.

Complete any of the following steps:

1. Display a Flow Action that is disqualified from bulk processing:
   a. In the navigation pane of Dev Studio, click **Records**.
   b. Expand the **Process** category, and then click **Flow Action**.
   c. Open the Flow Action that is missing from the **Select Action** list.
   d. On the **Action** tab of the Flow Action form, in the **Indicator** section, clear the **Disqualify this action from bulk processing** checkbox.
   e. Click **Save**.
2. Display Flow Actions for Cases:
   a. On the **Bulk actions** page, in the **Filter Property** column, remove the filter criteria that evaluate an Assignment.

©2025 Pegasystems Inc

> **For example:**
>
> Delete rows that contain Assigned to Operator or Assigned to Work Queue filter properties.

a. Click Filter Cases.
b. Select the checkbox next to each Case for which you want to run the Flow Action.

3. Display Flow Actions and connector Actions for Assignments:
   a. On the **Bulk actions** page, in the Filter Property column, add filter criteria that evaluate an Assignment.

> **For example:**
>
> Add rows that contain Assigned to Operator or Assigned to Work Queue filter properties.

   a. Click Filter Cases.
   b. Select the checkbox next to each Assignment for which you want to run the Flow Action.

Troubleshooting errors with Assignments stuck on pxAssignedOperatorID=pyBulkProcessing

Learn why Assignments get stuck during bulk operations and how to fix this issue to unblock Case processing.

Condition

When an agent or node goes down during bulk processing, Assignments, by default, are set to display in the *Assign-Worklist* of the *pyBulkProcessing* operator ID. However, the operator profile *pyBulkProcessing* does not exist, causing Cases to get stuck.

©2025 Pegasystems Inc

Cause: The default value for the *BulkProcessingID* Application Setting does not correspond to a valid operator profile

The default value for the *BulkProcessingID* Application Setting is *pyBulkProcessing*, but there is no corresponding operator profile for this value. As a result, Assignments cannot be processed when routed to this ID during exception scenarios.

> **TIP:**
>
> - Always configure the *BulkProcessingID* Application Setting to a valid operator profile based on your business needs.
> - Create a dedicated operator, for example *BulkProcessingAdmin*, to handle Assignments during exception scenarios.
> - Review and reassign Cases stuck under the *BulkProcessingAdmin* operator after resolving the exception.

Remedy

1. Override the Pega-delivered *BulkProcessingID* Application Setting Rule and save it into your application ruleset.

   Application Settings Rules are of the SysAdmin type. For more information, see Finding Rules by type, Creating a Rule specialized by a class or Ruleset, and Rules management.

2. In the newly-saved Application Setting Rule, replace the `BulkProcessingID` value with an existing or newly created operator profile ID, and save your change.

   For more information, see Defining values for an application setting.

©2025 Pegasystems Inc

NOTE: Ensure that you assign appropriate access privileges to the new operator. For more information, see Creating an operator ID and Defining user contact information and application access.

Result:

In exception scenarios, Assignments are automatically assigned to the new operator, and can then be reviewed and reassigned to the appropriate operator to unblock Case processing.

Creating ad hoc Assignments

Initiate one or more optional Assignments from the current Case by selecting a relevant operation in the **Actions** menu. By empowering users to create optional Assignments without navigating away from the current Case, you enhance your workflow effectiveness, convenience, and alignment with your business goals.

For example, in the process of evaluating a customer application for insurance services, the Case worker discovers certain risk factors that indicate a potential high-risk profile. In response, the Case worker makes an ad hoc decision to undertake the following actions:

- Initiate in-depth risk analysis.
- Collaborate with the underwriting team to assess the insurability of the applicant.

Before you begin:

Enable the creation of ad hoc Assignments in your workflow by adding the **Create Assignments** optional Action to your Case Type. For more information, see Adding optional Actions to Cases and Adding optional Actions to Stages.

©2025 Pegasystems Inc

Optionally, you can set a Service-Level Agreement for the ad hoc Assignments by extending the underlying *pySimpleCase* SLA Rule for this Action. For more information, see Setting SLAs for ad hoc Assignments.

IMPORTANT: As a best practice, prioritize regular Assignments over ad hoc Assignments in Case Management workflows. Regular Assignments provide structured processes, ensuring consistency and control, while ad hoc Assignments handle unexpected situations without altering the Case structure. For more information, see Collecting information from a user.

1. In the end user Portal, open a Case that you want to edit.
2. In the **Tools** section, click **Actions** > **Create Assignments**.

For example:

The following figure shows the creation of an ad hoc Assignment in an `Evaluate customer` Case:

Creating a new ad hoc Assignment from the current Case

3. In the **Create Assignments** dialog box, click **Add an Assignment**, and then specify the instructions, assignee, and deadline for the Assignment:
   a. In the **Instructions** field, provide the instructions for the assignee of the Assignment.
   b. In the **Assign to** field, select an assignee of the Assignment.

©2025 Pegasystems Inc

c. In the **Deadline** field, select the expected date for the Assignment completion.

For example:

The following figure shows instructions to initiate an in-depth risk analysis for a customer:

Defining a new ad hoc Assignment

4. To add more Assignments, repeat step 3.
5. Click **Check in**.

Result:

Your application displays the ad hoc Assignment in the Activities area of the Case. Users can add notes on completing the Assignment, as shown in the following figure:

Ad hoc Assignment created from the current Case

©2025 Pegasystems Inc

Related tasks

- Creating an Assignment in Pulse

Creating ad hoc Cases

You can handle business exceptions or track related tasks in your current Case by creating an ad hoc Case. Using an ad hoc Case, you can resolve exceptional processes without modifying the existing Case structure because ad hoc Cases are not instances of a Case Type.

For example, a Case worker, such as a human resources employee, can create an ad hoc Case when an application from a job candidate requires an additional background check.

IMPORTANT: Ad hoc Cases are not available in Constellation applications.

NOTE:

Pega Platform™ enables Case workers to create ad hoc Cases to manage unexpected and dynamic tasks. Ad hoc Cases, also known as dynamic Cases, are similar to Child Case Type, but instead of being run automatically, a user of the application may decide to run them based on their own decisions.

For example, when opening a bank account, the banker learns that the client opening the account has a high net worth and represents a lot of future business potential. The banker then makes an ad hoc decision that they should undertake the following actions:

- Send a hand-signed welcome letter from the branch president.
- Schedule a follow-up phone call in one week with the client to welcome them to the bank.
- Review the results.

©2025 Pegasystems Inc

Because ad hoc Cases are tracked in Pega Platform, if the bank finds that this is a common pattern for high net-worth clients, they should convert the ad hoc Case into a child Case that becomes a standard part of the Case Type for opening an account.

Before you begin:

Ensure that your dashboard includes the My Cases widget. For more information, see Adding a widget to your dashboard.

1. Log in to an end-user Portal.
2. In the **My Cases** widget, in the **Create an ad hoc Case** text field, enter a name for your ad hoc Case.

   For example:

   Enter `Additional background check`:

   Creating a new ad hoc Case

3. Click **Quick create**.
4. Add Assignments to the ad hoc Case:
   a. In the **My Cases** list, double-click the name of the newly created ad hoc Case.
   b. In the Case Summary pane, click **Actions**, and then select **Create Assignment**.
   c. In the **Ad hoc Case dashboard** section, click **Add an Assignment**, and then specify instructions for the Assignment, an assignee, and a deadline.

©2025 Pegasystems Inc

Adding Assignments to a new ad hoc Case

d. Click **Check in**.

Result:

Your application displays the Assignments in the **To do** list in the **Activities** area of the Case. Users can add notes upon completing the Assignment. If you create multiple Assignments in an ad hoc Case, your application displays the next Assignment after the user resolves the current Assignment.

- Accelerating Case processing with Pega GenAI, optional actions, and collaborative tools
- Designing the UI in Case Types
- Creating a child Case Type in Case Designer

Creating Cases from tables

Streamline operations by enabling users to initiate new Cases directly from a Case or landing page with a table. This way of creating Cases enhances user productivity, reduces navigation time, and ensures faster response times for Case Management.

©2025 Pegasystems Inc

NOTE: Only Constellation tables support adding new records.

Before you begin:

Enable adding new records:

- To enable the option for adding records in a list View with the table template in your Case Type, see Configuring tables.

  The following figure shows the checkbox for enabling the option in a list View:

©2025 Pegasystems Inc

Enabling the option for adding records in a list View with the table template in a Case Type

- To enable the option for adding records on landing pages with a list page template, see Configuring a list-based landing page.

  The following figure shows the checkbox for enabling the option in a list page:

Enabling the option for adding records on a landing page with the list page template

©2025 Pegasystems Inc

> **NOTE:** In a Case Type, depending on where you need your table, you can use a list View with the table template in a Tab or in a Step.
>
> For more information, see Creating Views, Creating Views for new Steps, and Working with Views.

1. Open a landing page or a Case that has the table with the option for adding new records.
2. In the table header, click the button for adding new records.
   - For a table in the Case UI, click the **Add** icon:

Adding records to a table in the Case UI

   - For a table on a landing page, click **Add**, as shown in the following figure:

©2025 Pegasystems Inc

Adding records to a table on a landing page

> **TIP:** The tooltip can be customized to display your own text. For more information, see Configuring tables on a landing page and Configuring tables.

3. Complete the **Create [Case name]** form with the Case details, and then click Submit.

> **Result:**
>
> Depending on where your table is, the new Case is accessible in the following ways:
>
> - For a table in the Case UI, the Case is immediately displayed at the top of the list of records and the system displays a pop-up notification with the new record details.
> - For a table on a landing page, the new Case immediately opens for processing. In the background, the system adds the new record at the top of the list of records in the table.

Related concepts

©2025 Pegasystems Inc

- Creating and managing Cases

Related tasks

- Creating Cases in bulk

Creating Cases from the navigation pane of a Portal

Create and process Cases in a Portal, so that users can perform their work more efficiently and meet their business goals. For example, a customer service representative can create a Case to collect and process information from a customer applying for a loan, and then resolve the Case to achieve a specific outcome, such as granting the loan.

A Case that a user creates is an instance of the specified Case Type, which is a predefined, reusable template of a business process that you want users of your application to follow. Case Types consist of Stages, Processes, and Steps, which a Case worker completes to resolve a Case.

Before you begin:

1. Create Case Types that you can reuse for multiple business processes. For more information, see Creating a Case Type manually and Creating Remote Case Types.
2. Define which Cases the user can create. For more information, see Organizing the main navigation for a standard work Portal

1. Log in to the Portal in which you process Cases.
2. In the navigation pane, click **Create**, and then select the Case Type that you want to use to create your new Case.

©2025 Pegasystems Inc

For example:

Creating a Case from a Home mortgage Case Type in User Portal

3. **Optional:** To discontinue work on a Case before submitting the Case for processing, click **Cancel**, and then select a relevant option:
   - To delete a Case, click **Delete**.

     NOTE: If you cancel the Case at the Create Stage, the Case enters the `Resolved-Cancelled` status. Worklist and Work Queue omit the Case.

     For more information, see Adding the Create Stage to existing Case Types and The Create Stage FAQ.

   - To close the Case and save your work, click **Save and close**.

©2025 Pegasystems Inc

Result:

You have created a Case, which can now be processed to achieve a specific result.

A new Home mortgage Case

- Case Types
- Remote Case Types
- The Create Stage
- Case Lifecycle elements
- Summary panel Views
- Accessing Views in your application
- Configuring Portals
- Organizing the main navigation for a standard work Portal

Creating Cases from the Actions menu of another Case

Start one or more Cases from the current Case by selecting a relevant operation from the Actions menu. By enabling users to create Cases on demand without navigating away from the main Case, you make your workflow more effective, convenient, and suited to your business goals.

©2025 Pegasystems Inc

For example, in an application to process company purchases, a user can start a new Case to create a new vendor, or update vendor data in the system.

Before you begin:

1. Add a supporting Process to your Case Type. For more information, see Adding supporting optional Processes to Cases.

   TIP: Give a meaningful name to the supporting Process, for example **Create a new vendor** because the name that you give to the Process is visible at run time as the optional Action name in the **Actions** menu.

2. Add the child Case to the **Create Case** Step of the supporting Process. For more information, see Creating a child Case Type in Case Designer.

   TIP: Keep your supporting Process simple without unnecessary Steps or Actions. The Process only needs the **Create Case** Step that triggers the relevant child Case.

1. Log in to an end-user Portal, for example, Web Portal.
2. Create or find a Case that you want to process.

TIP: You can locate a Case by searching for it or by looking in your Worklist or the **Recents** list.

For more information, see Creating Cases from the navigation pane of a Portal.

©2025 Pegasystems Inc

3. In the **Tools** section, click **Actions** > *[optional Action name]*.

For example:

Creating a new Case from the current Case

Result:

The system runs the selected child Case. If the new Case has the Create Stage with Steps configured, you can see a modal dialog box in which you can enter the new Case details, as in the following example:

Child Case created from the current Case

After you click **Submit**, the system saves the initial data to the new Case and displays a toast message with a link to the Case, giving you direct access to the new Case if you decide to immediately continue processing it.

©2025 Pegasystems Inc

Following Cases

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:

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.
2. Open a Case:
   - Search for the Case.
   - Click on the Case in the Worklist or Work Queue on your dashboard.
3. In the header of the Case, click the **Star** icon.
4. **Optional:** To unfollow the Case, click the **Star** icon again.

Result:

You now follow a Case and the utilities panel displays you as a follower of the Case:

Case followers in the utilities panel of a Case

- **Managing Case followers**

©2025 Pegasystems Inc

Related concepts

- Widgets in Constellation

Related tasks

- Enabling Case following

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.
2. Open a Case:
   - Search for the Case.
   - Click on the Case in the Worklist or Work Queue on your dashboard.
3. In the **Followers** section, click the **Manage followers** icon.

   For example:

   The Followers section in which you are the only follower

4. In the **Followers** dialog box, in the drop-down list, select one or more users in your application, and then click **Add**.

©2025 Pegasystems Inc

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.

Related tasks

- Enabling Case following

©2025 Pegasystems Inc

Interacting with parent and child Cases on multiple tabs

Understanding how to correctly handle parent and child Cases is crucial to maintaining data consistency and ensuring accurate Case processing, especially when interacting with parent and child Cases on multiple tabs within the same user session.

> ⚠ **IMPORTANT:** Applicable to traditional UI applications.

In Pega Platform™ Case Management, you can start multiple child Cases from a parent Case and work on the child Cases in parallel to the parent Case, which speeds up Case resolution.

Typically, you use child Cases when you want to model work that:

- Is performed by different parties
- Follows different reporting options
- Follows different time frames
- Is common to multiple Case Types

You add a child Case to a Case Type by adding the Create Case Automation Step to the Case Lifecycle. When encountering the Create Case Step, the system creates an instance of the specified child Case Type and copies data from the parent Case (cover Case), to the child Case (covered Case).

When a user works on both parent and child Cases in separate tabs within the same session, it's essential to understand the expected behavior. If a child Case is resolved, the parent Case's data on the Pega Clipboard is not automatically updated. When the user resolves the child Case and goes back to the parent Case, the child Case data is not updated on the parent Case Clipboard tool unless you specifically refresh the parent Case. To prevent committing stale data, you must refresh the parent Case tab before continuing with its processing.

PEGA ©2025 Pegasystems Inc

Related tasks

- Creating a child Case Type in Case Designer
- Sharing data between parent and child Cases

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 Constellation applications

Features	Description
Attachments	Supports adding files and links to Assignments. For more information, see Categorizing Case attachments.
Case history	Shows details about Assignments in the form of a table. For example, the information includes names of users who perform actions in an Assignment. The widget does not require additional configuration and is ready to use by default.
Case operator	Includes the name of the user who submits an Assignment. The widget does not require additional configuration and is ready to use by default.

©2025 Pegasystems Inc

Features	Description
Create	Begins a new Case, which is an instance of one of the Case Types that are defined in your application. For more information, see Creating Cases from the navigation pane of a Portal.
Dashboard	A Constellation landing page that uses the dashboard template. The Constellation dashboard component is a curated experience. Consequently, it is the application designers who specify what users see when they log in. Personalization at run time is not possible. For more information, see Configuring a dashboard.
Explore Data	Contains React-based reporting tools for Case Types. For more information, see Explore Data and Visualizing data with Insights .
Followers	Shows the list of users who track the progress of a certain Case. For more information, see Following Cases.
Home	Provides users with a quick overview of your application. You can populate the home page with out-of-the-box widgets, such as **Pulse**, **App announcements**, and **To do** list, as well

©2025 Pegasystems Inc

Features	Description
	as fields and Views that best meet the needs of your business. The Home page in a Constellation application is a curated experience. Consequently, the application designers specify what users see when they log in. For more information, see Home.
My Work	Provides quick access to Cases and Assignments of users.
Pulse	Provides users with a message feed in which to communicate with each other. The widget does not require additional configuration and is ready to use by default. For more information, see Collaborating with users by using Pulse and Configuring Pulse for Case Types.
Related cases	Shows the list of cases that are related to a certain Case. The widget requires additional configuration. For more information, see Configuring the Related Cases widget.
Search	Helps you control what data is available for the Search and Reporting Service (SRS).

©2025 Pegasystems Inc

Features	Description
	For more information, see Search landing page.
Stakeholders	Shows the list of users who are the first points of contact for a certain Case. The widget requires additional configuration. For more information, see Defining Case participants.
Tags	Lists tags that application users add to Cases. The widget requires additional configuration. For more information, see Configuring the Tags widget and Tagging in Cases.

For more information, see:

- Landing pages
- Working with Portals
- Configuring a Full Page View
- Widgets in Constellation

Out-of-the-box Portal features in traditional UI applications

Feature	Description
Create	Enables users to begin a new Case, which is an instance of one of the Case Types that is defined in your application and added to the **Create** menu.

©2025 Pegasystems Inc

Feature	Description
	For more information, see Creating Cases from the navigation pane of a Portal.
Dashboard	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.
Documents	Lists application documents that users create or follow. For more information, see Collaborating on shared content by using documents.
Explore Data	Contains React-based reporting tools for data and Case Types. For more information, see Explore Data and Insights.
Home	Provides users with a quick overview of the application. You can populate the home page with out-of-the-box widgets, such as **Pulse**, **App announcements**, and **To do** list, as well as fields and views that best meet the needs of your business. For more information, see Home.
My Work	Provides quick access to Cases and Assignments of the user.

©2025 Pegasystems Inc

Feature	Description
Pulse	Displays users' message feed from the **Pulse** gadget. For more information, see Configuring Pulse for Case Types and Collaborating with users by using Pulse.
Reports	Provides quick access to reports in the application. For more information, see Report Browser and Editing a report.
Search	Helps you control what data is available for the SRS. For more information, see Search and Reporting Service.
Spaces	Lists and helps users manage collaboration spaces. For more information, see Collaboration with users by using Spaces
Tags	Lists tags that application users add to Cases. For more information, see Tagging in Cases and Creating suggested tags for Cases.

For more information, see:

PEGA ©2025 Pegasystems Inc

- Custom pages
- Pages for applications
- Configuring Portals
- Organizing the contents of a Portal
- Organizing the main navigation for a standard work Portal
- Creating custom menus
- Summary panel Views

Reopening resolved Cases

Reopen a resolved Case if it was closed prematurely or erroneously. After reopening the Case, you can process and resolve the Case again.

> **Before you begin:**
>
> For applications that you update from Pega Platform™ 8.6 or earlier version, add and enable the *Reopen* Action in a Case Type. For more information, see Adding optional Actions to Cases and step 5 in Displaying optional Actions conditionally.

1. Log in to an end-user Portal.
2. Find and double click a resolved Case that you want to reopen.
   You can quickly locate a resolved Case by searching for it or by looking in your Worklist or the **Recents** list.
3. In the **Tools** section, click **Actions** > **Reopen**.
4. In the **Reopen** section, in the text field, provide the reason for reopening the Case.

©2025 Pegasystems Inc

For example:

Reopening a resolved Case

5. **Optional:** To move the Case to a selected Stage after reopening it, in the **change Stage** list, select the new Stage.
   After you reopen a Case, the Case Status changes to Open. If you do not supply a Stage to change to when you reopen the Case, it reopens in the first Stage after the Create Stage. Otherwise, it restarts the Stage that you selected.
6. Click **Submit**.

- Changing Case Statuses

Default solutions and extension points for the Reopen functionality

Pega Platform™ provides out-of-the-box options for reopening resolved Cases as well as extension points that you can use to extend the functionality.

Every Case has a Lifecycle, and at the end of its Lifecycle the Case is usually resolved. When a Case is resolved, the status of the Case is prefixed with `Resolved-`. A resolved Case is usually persisted in the system for a certain period of time before the Case is either archived or purged. However, when designing applications, there might be a requirement for users to be able to reopen and further process an already resolved Case.

©2025 Pegasystems Inc

For example, in a CRM application, a customer raises a dispute and a dispute Case is created in the system. However, the customer does not provide the details in time, and the dispute Case is marked as resolved. Later, the customer responds with their follow-up information, and the Case has to be reopened to be processed.

Default solutions for reopening resolved Cases

To handle scenarios in which a Case has to be reopened, Pega Platform provides the following out-of-the-box options for reopening resolved Cases:

- The Reopen Action in the Actions menu.
- The *reopenWorkObject* Process API.

The Reopen functionality works differently based on how the Case is defined:

- If the Case uses Stages, then the out-of-the-box functionality opens the Case and shows a Flow Action to move to the last Stage, with the option to change to other Stages in the Actions menu.
- If the Case does not use Stages, then the out-of-the-box functionality opens the Case and runs the Process identified by the *pyFlowName* property.

Extending the Reopen functionality

Best practice for customizing the Reopen functionality is to use the extension points provided by Pega Platform:

The *ReopenDefaults* activity
: This extension point activity is invoked in all scenarios in which a Case is reopened. For example, this extension point can be used to set the *pyFLowName* property to a different Process name and open a different Process, other than the default Process when the Case is reopened.

The *pyReopenToStage* Process
: This Process is invoked when the Case uses Stages.

Related concepts

©2025 Pegasystems Inc

- Flow Actions
- Stages in a Case Lifecycle

Related tasks

- Processes in a Case Lifecycle

Setting notification preferences

Control the type of notifications that you want to receive and select the notification channels over which you want to receive them.

For example, you can choose specific channels for receiving notifications, making it easier to relate to and manage your tasks effectively.

> **NOTE:** Only traditional UI applications support setting notification preferences at run time. For more information, see User experience.

> **IMPORTANT:** If the notification preference option is not displayed in the Portal that you use, ensure that the **Not configurable form UI** checkbox is inactive on the **Advanced** tab of the Notification that is configured for the **Notifications** widget of the Portal. For more information, see step 6.c in Configuring a Web gadget notification delivery Channel for Portal notifications.

Setting preferences for the general and Case Type-specific notifications

> **NOTE:** The general and Case Type-specific notification classification depends on how Pega Platform™ calls a Notification:
>
> **General**
>
> The system calls a Notification when Case-independent events occur, for example:

©2025 Pegasystems Inc

- A user mentions another user.
- A user likes a message that another user posted.

Case Type

The system calls a Notification when a Case-dependent event occurs, such as the posting of a Case comment.

> **NOTE:** 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 you can configure them by using the **Comments on Cases that I follow** option that is available in each Case Type. Additionally, you can also specify a different set of preferences for a specific Case that override the Case Type preferences. For more information, see steps 5 through 6.

1. In the navigation menu of the Portal that you use, click **Profile** > **Notification preferences**.

For example:

The following figure shows a sample **Profile** icon with the **Notification preferences** option:

Notification preferences option

©2025 Pegasystems Inc

2. If you want to continue receiving notifications, in the **Notification preferences** dialog box, ensure that the selection for the **Receive notifications** checkbox is active.
3. For each notification under the General or Case Type category, set your preference for each notification channel.

> **NOTE:** You define preferences for the notifications that you get directly in the **Notifications** widget in a Portal in the **Gadget** column.

> **For example:**
>
> The following figure shows a sample configuration of notification preferences for the general and Case Type-specific notifications:

©2025 Pegasystems Inc

Sample configuration of notification preferences

4. Click **Submit**.

Setting preferences for Case-specific notifications

> ⚠ **IMPORTANT:** If you disabled the **Receive notifications** option in step 2, the system stops all the notifications, including the ones that you set for a specific Case.

©2025 Pegasystems Inc

5. In the Actions header of a Case for which you want to change notification preferences, click **Actions** > **Manage notifications**.

For example:

The following figure shows the **Actions** menu of a Case with the **Manage notifications** option:

Actions menu with the Manage notifications option

6. Set preferences for the current Case by selecting the **Override at instance level** checkbox, and then in the notifications list, update your preferences.

NOTE: You define preferences for the notifications that you get directly in the **Notifications** widget in a Portal in the **Gadget** column.

7. Click **Submit**.

- Engaging and notifying stakeholders
- Configuring Pulse email notifications
- Notifying Case participants about Case events
- Engaging mobile users with push notifications

©2025 Pegasystems Inc

Tagging in Cases

You can use tags to classify, manage, and report on tagged Cases in a preferred way. Using tags promotes flexibility and collaborative Case Management.

For example, in a financial application, you can create the tags `premium` and `loan` to flag Cases related to specific types of accounts, or tags to flag specific types of issues, for example, `customer_satisfaction`:

The Tags widget showing tags that are associated with a Case

You can create an initial list of tags for your application to provide end users with tagging suggestions:

Enabling tags for a Case Type and creating the initial set of tags

The list helps end users find and apply the most often used tags and promotes consistent tagging throughout the application. For more information, see Creating suggested tags for Cases.

At run time, you can tag Cases by using existing tags or by adding new tags:

PEGA ©2025 Pegasystems Inc

Selecting tags from different categories

You can tag a Case regardless of the current Case Status. For more information, see Adding tags to Cases at run time.

You can gather relevant information faster by filtering tagged Cases in several ways, for example, by recently added, recently viewed, or most frequently used tags:

Filtering Cases with a selected tag

©2025 Pegasystems Inc

For more information, see Filtering tagged Cases.

- Enabling Case tags
- Configuring the Tags widget
- Tagging a message in Pulse
- Collaborating with users by using Pulse

Adding tags to Cases at run time

Find relevant information faster and enhance collaboration on Cases by associating tags with Cases. For example, in an insurance application, you can add tags such as `premium` or `high_risk` to convey vital information about a Case.

> **Before you begin:**
>
> 1. Enable Case tags so that tagging is available at run time. For more information, see Enabling Case tags.
> 2. Add the Tag custom page to the navigation menu of the end-user Portal, to enable searching for tags on the Tag landing page. For more information, see Organizing the main navigation for a standard work Portal and Custom pages.

At run time, you can add tags to a Case by selecting from a list of available tags or by creating new tags. For more information about creating a set of default suggested tags for Cases, see Creating suggested tags for Cases.

You can tag a Case regardless of its current Case Status.

1. Log in to an end-user Portal.
2. Open a Case:
   - Search for the Case.
   - Click on the Case in the Worklist or Work Queue on your dashboard.
3. In the Summary pane of the Case, click **Actions** > **Manage tags**.

©2025 Pegasystems Inc

4. In the **Manage Tags** dialog box, select a tag from the list of existing tags or enter a new tag name, for example `Reports`.

> **NOTE:** Tags cannot contain spaces and are limited to 64 characters.

> **TIP:** You can also add tags by clicking items in the **Tags recently used by me** section of the dialog box.

5. Click **Add**.
6. **Optional:** To add more tags, repeat steps 4 to 5.
7. **Optional:** To remove a tag association from a Case, in the **Manage Tags** dialog box, click the **Unlink this tag** icon next to the tag that you want to remove.
8. Close the dialog box.

Result:

The tags that you add appears in the Tags widget, in the associated tags list:

List of tags associated with your Case

> **NOTE:** You can now manage Case tags by clicking the **Manage tags** icon, and then adding or removing tags.

©2025 Pegasystems Inc

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.

Filtering tagged Cases

Use tags to search, access, and report on Cases faster. You can filter tagged Cases in several ways, for example, by viewing only the recently added or most frequently used tags. To find specific information, you can also create specific tags and then search Cases with those tags. For example, you can look for Cases with the `premium` tag to identify your key clients or accounts.

Before you begin:

1. Enable Case tags so that tagging is available at run time. For more information, see Enabling Case tags.
2. Add the Tag custom page to the navigation menu of the end-user Portal to enable searching for tags on the Tag landing page. For more information, see Organizing the main navigation for a standard work Portal and Custom pages.

1. Log in to an end-user Portal.
2. In the navigation pane on the left, click **Tags**.
3. Search for Cases in one of the following ways:
   - To filter Cases by a specific tag, enter the tag name in the **Search** field, and then select the tags from the **Search results** section.
   - To filter Cases by tags that you added recently, select the tags from the **Recently added by me** section.
   - To filter Cases by tags that any other user added recently, select the tags from the **Recently added by anyone** section.

©2025 Pegasystems Inc

- To apply more out-of-the-box filters that categorize tags, click the **Filter** icon next to the **Search** field, and then select any of the available options, for example, **Recently viewed by me**.

Result:

The **Selected tags** section displays Cases that use the selected tags. You can now preview the Cases by clicking the Case names.

> **NOTE:** If your tags also appear in any Pulse messages, the search results provide this information on the **Posts** tab.

Filtering Cases by the `premium` *tag*

4. **Optional:** To add more tags to filter Cases that use any of the selected tags, in the **Selected tags** section, click **Edit** to enter or select more tags in the search box, and then click **Apply**.

Creating suggested tags for Cases

Improve Case search and ensure that users apply consistent tagging by creating suggested tags. When users apply consistent tagging in Cases, they obtain information that they need faster. By defining sets of suggested tags you shorten Case processing time as users can select tags from a list when they tag a Case instead of typing tags manually.

©2025 Pegasystems Inc

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. Click the **Settings** tab, and then, in the navigation pane, click **Collaboration**.
3. In the **Tags** section, in the **Configure tags** field, enter a tag that you want to create as a suggested tag, and then click **Add**.

NOTE: Tags cannot contain spaces and are limited to 64 characters.

Tags can use underscores, so a tag with multiple words can be written as `customer_satisfaction`.

For example:

The following figure shows tags that are now in the list of suggested tags for every Case in a given Case Type:

©2025 Pegasystems Inc

Suggested tags for a Case Type

4. **Optional:** To remove a tag from the list of default tags associated with a Case Type, click **Remove** next to the tag.
5. Click **Save**.

Result:

At run time, when users tag a Case, they can select the suggested tags, instead of entering tags manually:

©2025 Pegasystems Inc

A Case with a list of suggested tags

Transferring Assignments

You can transfer an Assignment from one user to another. For example, you can transfer an Assignment to a team member with more experience or less workload. By reassigning work to other members on your team, you can resolve Cases more quickly and effectively.

The **Transfer Assignment** optional Action enables users to manually transfer an Assignment to another user, a Work Queue, or team.

Before you begin:

Enable transfer of Assignments for your process by adding the **Transfer Assignment** optional Action to your Case Type. For more information, see Adding optional Actions to Cases and Adding optional Actions to Stages.

1. Log in to an end-user Portal.
2. Find and double click the Case that you want to open.

©2025 Pegasystems Inc

3. In the **Tools** section, click **Actions** > **Transfer Assignment**.
4. In the **Activities** pane of the Case, in the **Transfer Assignment** section, select the new assignee.

For example:

You can select **User**, and then enter the user name, as shown in the following figure. After the transfer, the Assignment appears in the Worklist of the new assignee.

An example Case with a new user selected to process the Case

5. Click **Transfer**.

Insights in Case Management

Pega Platform™ provides a convenient way to retrieve specific application data and display the information in the form of comprehensive table-based Insights or visually appealing chart-based Insights. Customizing your landing pages and Case Types with comprehensive visual data representations helps Case workers to better understand trends and make informed decisions. Actionable data items in Insights also help streamline decision-making processes and improve work processing efficiency.

©2025 Pegasystems Inc

Insights on landing pages

Insights are accessible in the **Explore Data** feature, which can use either Constellation or hybrid traditional environments. As a result, you can add Insights to the landing pages of applications. For more information, see Constellation UI and Explore Data.

You can create a landing page that displays an Insight with Cases. For example, when authoring the Work Portal, you can add a landing page to feature an Insight with data related to orders, as shown in the following figure:

UI authoring of a Portal

The figure includes the following areas:

1. List of landing pages in a Portal configuration pane.
2. Insight added to a region of a selected landing page in the landing page configuration view.

The following figure shows the landing page at run time:

©2025 Pegasystems Inc

Landing page with an Insight

In an Insight, you can perform several actions, such as grouping, filtering, and sorting Cases. For more information, see Run-time behavior of Constellation tables.

For more information about configuring an Insight for a landing page, see Adding Insights to your application.

Insights in Cases

In Constellation or Theme Cosmos environments, you can include a custom Insight in a full Page Case View of your Case Type.

For example, when authoring an Order Case Type, you can edit the **Details** tab to feature a chart-based Insight with visualized orders of specific products in the last quarter.

In the following example, the **Details** tab features an Insight with data related to orders:

©2025 Pegasystems Inc

UI authoring of a Case Type

The figure includes the following areas:

1. Case Type summary panel with tabs.
2. Sample chart-based Insight displayed in the open tab content area.

For more information about configuring an Insight for a Case Type, see Creating Insights from UI authoring.

Related concepts

- Insights and Reporting

Related tasks

- Visualizing questionnaire data by creating Insights in Constellation

©2025 Pegasystems Inc

Collaborating on Cases

Complete your Microjourney faster by providing collaboration tools for your application users, such as customer service representative (CSRs). Through a transparent exchange of messages in an open discussion, customer service representatives (CSRs) can resolve Cases faster and with better effect.

Application users can collaborate by posting Pulse messages, maintaining spaces, and working on documents.

For relevant training materials, see the Collaboration with users module and Collaborating with Pulse and Tags challenge on Pega Academy.

- **Collaborating with users by using Pulse**
- **Collaboration with users by using Spaces**
- **Collaborating on shared content by using documents**

Collaborating with users by using Pulse

Resolve your work faster by using Pulse. By sharing information in Pulse, users can work together to complete their work more effectively.

- **Posting a message in Pulse**
- **Replying to a message in Pulse**
- **Restricting Case references in Pulse**
- **Tagging a message in Pulse**
- **Managing Pulse messages**
- **Creating an Assignment in Pulse**

PEGA ©2025 Pegasystems Inc

Related concepts

- Preparing for collaboration with users by using Pulse

Related tasks

- Configuring Pulse for Case Types

Posting a message in Pulse

Post a message in Pulse to share information, for example, the status of a Case, or to ask a question in a conversation thread. By posting your message in the correct context, you control who can receive notifications, view your messages, and post replies.

For example, users who are preparing a presentation for a customer can exchange ideas in a Pulse conversation, which helps keep the discussion within the context of a single user story.

For relevant training materials, see the Collaboration tools in App Studio article on Pega Academy.

> **Before you begin:**
>
> The option to post a Task in Pulse is available only in the global Pulse feed and spaces. To enable posting a Task in a particular context, configure the feed gadget. For more information, see Enabling users to post messages in the activity feed.

1. Choose the context for your message, based on your target audience and message content:

©2025 Pegasystems Inc

Choices	Actions
Application developers and stakeholders	In the header of App Studio, click the Toggle Developer collaboration icon.
Case followers and participants	Open a Case by searching for it or by reviewing the Worklists and Work Queues on your dashboard.
Members of a team	a. In the navigation pane of the Case Manager Portal, click My Teams. b. Click the name of the team to which you want to address your message.
A specific user	a. In the navigation pane of the Case Manager Portal, click My Teams. b. Click a team name. c. In the **Members** section, click Edit Members. d. In the **My team** list, click the name of the user to whom you want to address your message.

2. Add a message:
   - To add a message for all users within this context, click Post > Post.
   - To add a message for specific users within this context, click Post > Private post, press the Down arrow key, and then select the message recipients.

     > **NOTE:** All the users involved in a private conversation can add more users to the conversation by referencing them, but only the owner can remove users from the conversation.

   - To create and assign a Task in Pulse, click Post > Task.

©2025 Pegasystems Inc

For more information, see Creating a Task in Pulse.

3. Click **Start a conversation**, and then enter your message text.
4. **Optional:** Enhance your message by controlling its presentation and the information that supports it:

Choices	Actions
Personalize the look and feel of your message	a. Click the **Formatting help** icon. **NOTE:** Constellation applications do not have the **Formatting help** icon, so you can apply the formatting directly in the post. **Result:** The system displays the syntax that you can use to format your text in different ways:

©2025 Pegasystems Inc

Choices	Actions

```
Style
  **Bold** or __Bold__
  _Italic_ or *Italic*
Images
  ![Alt Text](url)
Links
  [Text](url)
Lists
  1. Item 1
  2. Item 2
  * Bullet Point 1
  * Bullet point 2
Headings
  # Heading 1
  ## Heading 2
```

Enhancing a Pulse message with formatting

For more information, see Markdown in Pulse.

b. Apply a syntax from the Style, Lists, or **Headings** section to the relevant text in your message.

©2025 Pegasystems Inc

Choices	Actions
	For example: You can apply the bold style and italics to emphasize key information, and use headings and lists to better organize the message, as in the following figure: *Pulse message with formatting*
Add links	a. Click the **Formatting help** icon. **NOTE:** Constellation applications do not have the **Formatting help** icon, so you can apply the formatting directly in the post. For more information, see Markdown in Pulse. b. Apply a syntax from the **Images** or **Links** section to the URL in your message text. **For example:** To add a link to the main page of the English Wikipedia, enter `[Wikipedia - The Free Encyclopedia]` `(https://en.wikipedia.org)`.

©2025 Pegasystems Inc

Choices	Actions
	 Pulse message with a link
Provide additional information by adding an attachment	a. Click the Add attachment icon. **NOTE:** In Constellation applications, the icon name is Attach files. b. Upload a file from your local system or an external location. **Result:** The system associates the attachment with your message and the context in which you post the message, for example, a Case. *Attachments embedded in a Pulse message*
Reference a user, space, document, or Case	**NOTE:** Only traditional UI Pega Platform applications fully support this feature. Applications that you build on Constellation support referencing only users. a. Place your cursor in the message text, and then enter @.

©2025 Pegasystems Inc

Choices	Actions
	b. Select whether you want to reference a user, space, document, or Case. c. Type one or more characters to narrow the list of results, and then select a value. **Result:** Individual users and users of the spaces that you reference receive notifications about your message, based on their notification settings. Documents and Cases that you reference become links in your message. *Referencing a user and a document in a Pulse message* d. **Optional:** To help users create messages more quickly, limit the list of Cases that they can reference in a message. For more information, see Restricting Case references in Pulse.
Add tags to your message	**NOTE:** Only traditional UI Pega Platform applications fully support this feature. In applications that you build on Constellation, you can also create tags, but clicking a tag does not open a **Tags** landing page. a. Place the cursor in your message text, and then enter #. b. Type one or more characters, and then select a tag from the list of matching results.

©2025 Pegasystems Inc

Choices	Actions
	Result: If there are no matches, the application creates a new tag from the text that you enter. c. **Optional:** To add another tag, enter a blank space, and then repeat step 4.b. **For example:** *Multiple tags in a Pulse message*

5. Click Post.

- Replying to a message in Pulse
- Extension points and supporting Rules for attachments
- Preparing for collaboration with users by using Pulse
- Configuring the display of feed sources

Replying to a message in Pulse

You can reply to a message in Pulse to continue a conversation. For example, a project manager responds to a message in which one of the developers asks for clarifications about a task in the upcoming sprint.

When you reply to another user's reply, the application automatically adds the name of the user at the beginning of your comment.

1. Choose the context for your message, based on your target audience and message content:

©2025 Pegasystems Inc

Choices	Actions
Application developers and stakeholders	In the header of App Studio, click the **Toggle Developer collaboration** icon.
Case followers and participants	Open a Case by searching for it or by reviewing the Worklists and Work Queues on your dashboard.
Members of a team	a. In the navigation pane of the Case Manager Portal, click **My Teams**. b. Click the name of the team to which you want to address your message.
A specific user	a. In the navigation pane of the Case Manager Portal, click **My Teams**. b. Click a team name. c. In the **Members** section, click **Edit Members**. d. In the **My team** list, click the name of the user to whom you want to address your message.

2. Find the relevant message in the Pulse conversation.
3. **Optional:** To acknowledge the message without sending a reply, click the **Like** icon.
4. Click **Comment**, and then enter your message text.
5. **Optional:** Enhance your message by controlling its presentation and the information that supports it:

Choices	Actions
Personalize the look and	a. Click the **Formatting help** icon.

©2025 Pegasystems Inc

Choices	Actions
feel of your message	**NOTE:** Constellation applications do not have the **Formatting help** icon, so you can apply the formatting directly in the post. **Result:** The system displays the syntax that you can use to format your text in different ways:

©2025 Pegasystems Inc

Choices	Actions
	Style **Bold** or __Bold__ _Italic_ or *Italic* Images ![Alt Text](url) Links [Text](url) Lists 1. Item 1 2. Item 2 * Bullet Point 1 * Bullet point 2 Headings # Heading 1 ## Heading 2 *Enhancing a Pulse message with formatting* For more information, see Markdown in Pulse. b. Apply a syntax from the Style, Lists, or **Headings** section to the relevant text in your message.

©2025 Pegasystems Inc

Choices	Actions
	For example: You can apply the bold style and italics to emphasize key information, and use headings and lists to better organize the message, as in the following figure: *Pulse message with formatting*
Add links	a. Click the **Formatting help** icon. **NOTE:** Constellation applications do not have the **Formatting help** icon, so you can apply the formatting directly in the post. For more information, see Markdown in Pulse. b. Apply a syntax from the **Images** or **Links** section to the URL in your message text. **For example:** To add a link to the main page of the English Wikipedia, enter `[Wikipedia - The Free Encyclopedia]` `(https://en.wikipedia.org)`.

©2025 Pegasystems Inc

Choices	Actions
	 Pulse message with a link
Provide additional information by adding an attachment	a. Click the Add attachment icon. **NOTE:** In Constellation applications, the icon name is Attach files. b. Upload a file from your local system or an external location. **Result:** The system associates the attachment with your message and the context in which you post the message, for example, a Case. *Attachments embedded in a Pulse message*
Reference a user, space, document, or Case	**NOTE:** Only traditional UI Pega Platform applications fully support this feature. Applications that you build on Constellation support referencing only users. a. Place your cursor in the message text, and then enter @.

©2025 Pegasystems Inc

Choices	Actions
	b. Select whether you want to reference a user, space, document, or Case. c. Type one or more characters to narrow the list of results, and then select a value. **Result:** Individual users and users of the spaces that you reference receive notifications about your message, based on their notification settings. Documents and Cases that you reference become links in your message. *Referencing a user and a document in a Pulse message* d. **Optional:** To help users create messages more quickly, limit the list of Cases that they can reference in a message. For more information, see Restricting Case references in Pulse.
Add tags to your message	**NOTE:** Only traditional UI Pega Platform applications fully support this feature. In applications that you build on Constellation, you can also create tags, but clicking a tag does not open a **Tags** landing page. a. Place the cursor in your message text, and then enter #. b. Type one or more characters, and then select a tag from the list of matching results.

©2025 Pegasystems Inc

Choices	Actions
	Result: If there are no matches, the application creates a new tag from the text that you enter. c. **Optional:** To add another tag, enter a blank space, and then repeat step 5.b. **For example:** *Multiple tags in a Pulse message*

6. Click Post.

Result:

Users receive notifications about your reply and any attachments that you provide, based on their notification preferences.

- Setting notification preferences
- Posting a message in Pulse
- Preparing for collaboration with users by using Pulse

Restricting Case references in Pulse

You can control which cases users can reference in the messages that they post to Pulse. By limiting the choices in the list of suggested Cases, you can help users create messages more quickly.

©2025 Pegasystems Inc

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Reports** category, and then click **Report definition**.
3. Open the *pyGetCasesForReferencesFromIndex* report definition that applies to the *Work-* class.
4. Copy the report definition to an open Ruleset in your application. For more information, see Copying a Rule or data instance.
5. Create a filter Condition that excludes Cases or instances of a specific Case Type.
   a. On the **Query** tab, in the **Edit filters** section, click **Add filter**.

      > **Result:**
      >
      > Your application populates the **Condition** field and adds its value to the Boolean Expression in the **Filter Conditions to apply** field.

   b. In the **Column source** field, enter `.pxObjClass`.
   c. In the **Relationship** list, select **Is not equal**.
   d. In the **Value** field, enter the class of the Case Type to exclude.
   e. **Optional:** To exclude more Case Types, click **Select values**.
6. Click **Save**.

- Posting a message in Pulse
- Finding Rules by class
- Preparing for collaboration with users by using Pulse

Tagging a message in Pulse

Categorize a message by tagging it in Pulse. By associating similar messages, you can help users find information that they need to resolve a Case. For example, if all Pulse comments that are related to a bug contain the bug ID tag, users who work on the bug can quickly find all of the related information.

©2025 Pegasystems Inc

> **NOTE:** Only traditional UI Pega Platform applications support adding tags to existing messages. Applications that you build on Constellation do not support this feature.

The following guidance applies to Case messages, however, you can communicate in other contexts, such as a team or application. For more information, see Posting a message in Pulse.

1. Open a Case:
   - Search for the Case.
   - Click on the Case in the Worklist or Work Queue on your dashboard.
2. Hover over a message.
3. Click the **Show more** icon, and then click **Add tag**.
4. Add a tag to the message:
   - To create a new tag, enter a unique name in the text field.
   - To reuse a tag, press the Down arrow key in the text field, and then select a tag.
5. Click **Add**.
6. Click **OK**.

- Posting a message in Pulse
- Preparing for collaboration with users by using Pulse
- Tagging in Cases

Managing Pulse messages

Actively manage messages in Pulse so that you can quickly find relevant conversations or attachments.

Use the following techniques to manage your messages in Pulse:

- Collaboration with users by using Spaces
- Preparing for collaboration with users by using Pulse

©2025 Pegasystems Inc

Viewing your activity feed

You can view your activity feed to find Pulse messages that you post or bookmark, and messages that are related to you. By filtering information from an aggregate source, you can find a specific conversation without opening individual Cases.

1. In the navigation pane, click **Pulse**.
2. **Optional:** To find a message more quickly, filter the results.
   a. Click the **Filter** icon.
   b. Clear the checkbox next to a type of message to exclude it from the results.
   c. Click **Apply**.
3. Review the list of results to find your message.
4. **Optional:** To continue the conversation for this message, click **Comment**.

- Replying to a message in Pulse

Bookmarking a message in Pulse

You can bookmark a message in Pulse that you consider important, so that you can easily find the message later in your activity feed.

The following guidance applies to Case messages, however, you can communicate in other contexts, such as a team or application. For more information, see Posting a message in Pulse.

1. Open a Case:
   - Search for the Case.
   - Click on the Case in the Worklist or Work Queue on your dashboard.
2. Hover over the message, and then click the **Show more** icon.
3. Click **Bookmark**.

- Removing a bookmark in Pulse
- Posting a message in Pulse
- Viewing your activity feed

©2025 Pegasystems Inc

Removing a bookmark in Pulse

You can remove a bookmark in Pulse when a message is no longer relevant to you.

1. In the navigation pane, click **Pulse** to open your activity feed.
2. Click the **Filter** icon.
3. Select the **My bookmarked messages** checkbox.
4. Click **Apply**.
5. Hover over a message, and then click the **Show more** icon.
6. Click **Remove Bookmark**.

- Bookmarking a message in Pulse
- Posting a message in Pulse
- Viewing your activity feed

Following messages from a specific user

Follow users to view the posts that they make in your Pulse activity feed.

> **Before you begin:**
>
> Ensure that your dashboard includes the Pulse activity feed widget. For more information, see Adding a widget.

1. Open the user profile for the user that you want to follow.
   For example, you can open the profile of a team member by completing the following steps:
   a. Click **My Teams**.
   b. Click a team name.
   c. In the **Members** section, click the name of the user.
2. Click **Follow**.

©2025 Pegasystems Inc

Result:

Your user profile displays the users whom you follow and the users who are following you. An icon indicates whether a user is online.

- Collaboration with users by using Spaces

Unfollowing messages from a specific user

Unfollow users so that you no longer see their posts in your Pulse activity feed.

TIP: If you have the user's profile open, you can quickly unfollow the user by clicking **Following**.

1. From your Profile menu, select **Profile**.
2. In the **Following** section, click **See all**.
3. Click the **Checkbox** icon for each user that you want to unfollow.
4. Click **Submit**.

- Following messages from a specific user

Deleting a message in Pulse

You can delete a message in Pulse when it is no longer relevant to a conversation.

1. Find the message in your activity feed.
2. Hover over the message, and then click the **Show more** icon.
3. Click **Delete**.

- Posting a message in Pulse
- Replying to a message in Pulse

©2025 Pegasystems Inc

Creating an Assignment in Pulse

To manage Case work that needs to be completed within a specific period of time, create a Task in Pulse and assign the Task to other Case users or to yourself. For example, you can assign a task to yourself as a reminder to upload address documents to a car loans Case by the end of the day.

Before you begin:

In Dev Studio, enable the **Task** post type.

For more information, see Enabling users to post messages in the activity feed.

1. Open a Case:
   - Search for the Case.
   - Click on the Case in the Worklist or Work Queue on your dashboard.
2. Click **Post** > **Task**.
3. Enter a name for the Assignment.
4. In the **Assignee** field, press the Down Arrow key, and then select a user to assign the Assignment.
5. In the **Due date** field, select the date and time by which the Assignment should be completed.
6. **Optional:** In the **Additional details** box, enter more information about the Assignment.
7. Click **Create**.

Result:

Your Pulse interface and the Pulse interface of the assignee display the Assignment. If the assignee does not complete the Assignment, they receive two reminder notifications based on their notification preferences. The first

©2025 Pegasystems Inc

notification arrives when half of the time till the due date elapses, and the second one appears after the deadline.

For example:

If the due date is in 24 hours, the assignee receives the first notification after 12 hours and the second one after 24 hours.

What to do next:

The assignee can click the **Complete** icon when they finish work on the Assignment. The assignee and you can comment on and like the Assignment. You can edit the Assignment when it is still open, and reopen the Assignment after its completion.

- Posting a message in Pulse
- Preparing for collaboration with users by using Pulse

Collaboration with users by using Spaces

By creating spaces, you gather users that are professionally connected within a single digital community, so that they can collaborate on specific areas of interest. When you restrict the discussion of certain topics to the users of a space, you avoid broadcasting irrelevant messages to all users in your application and create more effective communication.

For example, you are a team leader who wants to hire new members for your team. To discuss prospective candidates with management, you can create a Hiring space that includes the manager, senior manager, and director of the team.

NOTE: The feature is not available in Constellation applications. You can only configure an application to launch the feature in a section-based UI in a new

©2025 Pegasystems Inc

> browser tab. For more information, see Adding traditional landing pages to the navigation menu and Launching traditional pages from Constellation portals.

Spaces are areas in the application that enable groups of users to collaborate on a specific area of interest. In spaces, users can collaborate and exchange information by posting Pulse messages, uploading attachments, linking documents, or by pinning Cases, documents, reports, and other spaces.

To distribute work, plan outcomes, and ensure that every space member is on track, you can create a Task board, and then manage tasks in a comprehensible and clear way, even if the members of a space belong to different teams. For instance, you can populate a Task board that is visible to managers and members of various teams that cooperate on a joint project.

To make your content more relevant, create a sub-space that gathers users within a space. For example, you can create a sub-space for customer service representatives (CSRs) that work together on a particular project, so that you do not involve the whole team.

The following images show the Spaces landing page and the collaboration interface for a space:

©2025 Pegasystems Inc

The Spaces landing page by default displays the spaces of which the user is a member

The collaboration interface for a space

©2025 Pegasystems Inc

The following tasks can help you share information with other users by using spaces:

- Collaborating with users by using Pulse
- My Pega: My Spaces (video)
- Spaces overview (video)

Creating a space

Create a space to collaborate with users in your application on a specific area of interest, for example, to discuss training for new hires in your team.

1. In the header of App Studio, navigate to a Portal that contains the **Spaces** widget.
2. In the navigation pane, click Spaces.
3. Click Create space.

> **NOTE:** The user who creates a space becomes the owner of the space.

4. Enter a name for the space.
5. **Optional:** To give a short description to the space, enter some relevant text in the Description field.
6. In the Space type menu, select who can see and access the space.
   - Public – All users in your application can see and join the space. The owner can also invite users to the space.
   - Private – All users in your application can see the space and request access. The owner can also invite users to the space.
   - Unlisted – Only the owner and users who are invited by the owner to join the space can see it.
7. **Optional:** To create tasks in Pulse and use a Task board in your space, select the Enable task tracking checkbox.
   You can also enable task tracking after you create a space by clicking Edit space on the Activity tab of your space.

©2025 Pegasystems Inc

For more information, see Creating and managing tasks in a space and Creating an Assignment in Pulse.

8. **Optional:** To associate an image with the space, in the **Featured image** section, click **Update space image**, and then browse to and upload an image.

For example:

A new space setting with a custom image added to the space

Result:

The image is visible in the space preview on the Spaces landing page and when you open the space.

9. Click **Done**.

©2025 Pegasystems Inc

Result:

A space is created with a Pulse interface for discussions and sections that contain the space details, as shown in the following image:

A newly created space interface

The newly created space is visible on the Spaces landing page, as shown in the following image:

The Spaces landing space with the newly created space

©2025 Pegasystems Inc

Joining a space

Join a space to discuss a topic of specific interest with other members, for example, to exchange views about prospective candidates for your team.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Join a selected space by doing one of the following actions:
   - To join a public space, in the space preview, click **Join**.

     You can now instantly collaborate with members of the space.

   - To join a private space, in the space preview, click **Request access**.

     When the owner approves the request, you receive a notification, after which you can collaborate with members of the space.

   - To join an unlisted space, wait for an invitation from the owner of the space.

     When the owner invites you, you receive a notification, after which you can collaborate with members of the space.

> **For example:**
>
> The following images show previews of a public space (left), and a private space (right) with options to join and request access:

©2025 Pegasystems Inc

Joining a public space and requesting access to a private space

TIP: To leave a space, click Actions > Leave.

Communicating with members of a space

Communicate with the members of a space by using Pulse to discuss relevant topics, for example, training for new hires.

Before you begin:

Ensure that you have owner or member access to a space:

- Create a space to become the owner. For more information, see Creating a space.
- The owner of the space sets you as the owner. For more information, see Updating the owner of a space.

©2025 Pegasystems Inc

- Join the space to become a member. For more information, see Joining a space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space in which you want to start or continue a discussion.
4. On the **Activity** tab, use Pulse to communicate with members of the space.

   For more information, see Posting a message in Pulse, Replying to a message in Pulse, and Searching for Pulse messages in spaces.

   TIP: Select the types of notifications that you receive for the space by clicking **Actions** > **Notification preferences** and selecting the **Override at instance level** checkbox.

5. **Optional:** To add content to support your discussions, upload or pin files, URLs, documents, or cases and other spaces in your application.

   For more information, see Adding content to a space and Pinning content to a space.

6. If you are the owner of the space, moderate discussions and content by deleting messages, attachments, and pins that are irrelevant or inappropriate.

   For more information, see Managing Pulse messages.

PEGA ©2025 Pegasystems Inc

Result:

Pulse message with an attachment posted in a space

Pinning content to a space

Communicate with members of a space more effectively by pinning content that supports discussion in that space. For example, you can pin the job profiles of prospective candidates to a Hiring space.

A pin is a reference to a file, URL, document, Case, or space. You can add and edit titles and images for pins, search for pins, filter pins by pin type, and 'like' pins. You can also pin existing content in one space to another space.

Before you begin:

Ensure that you have owner or member access to a space:

- Create a space to become the owner. For more information, see Creating a space.
- The owner of the space sets you as the owner. For more information, see Updating the owner of a space.
- Join the space to become a member. For more information, see Joining a space.

©2025 Pegasystems Inc

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space to which you want to pin content.
4. On the **Board** tab, click **Add pin**.
5. In the **Create pin** modal dialog box, select a pin type.
6. Select the type of content to pin to the space:
   - For the **Case**, **Document**, or **Space** pin types, press the Down arrow key, and then choose a value from the field that appears, or upload a file.
   - For the **URL** pin type, enter a URL in the field that appears.
7. **Optional:** Update the pin title and image.

   > **Result:**
   >
   > The title and image appear in the pin preview on the **Board** tab.

8. Click **Pin to space**.
9. **Optional:** To pin the current space to another space, click **Actions** > **Pin to space**, and then choose or create the spaces to which you want to pin the current space.
10. **Optional:** To pin the content to another space, perform the following actions:
    a. Hover over the pin, and then click the **Show more** icon.
    b. Click **Repin**.
    c. In the modal dialog box, choose or create the spaces to which you want to pin the content.

©2025 Pegasystems Inc

Result:

The content that you pin appears on the **Board** tab of a space, which provides a comprehensive overview of the content that you link to the space.

TIP: You can reorganize pins to place the most relevant and the most important pins at the top of the list.

For example:

Pins with links to the candidates' profiles pinned to the board of the space:

©2025 Pegasystems Inc

Space board with links to candidates' profiles

Editing a pin in a space

Stimulate effective communication by updating the title and image of a pin, or adding an image to a pin to replace the default image. For example, you can add candidates' photos to the pins with job profiles in a Hiring space, to help the space members identify individual candidates.

Before you begin:

Ensure that you have owner or member access to the space:

©2025 Pegasystems Inc

- Create a space to become the owner. For more information, see Creating a space.
- The owner of the space sets you as the owner. For more information, see Updating the owner of a space.
- Join the space to become a member. For more information, see Joining a space.

NOTE: If you are a member of a space, you can only edit pins that you create.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space in which you want to edit a pin.
4. On the **Board** tab, search for the pin that you want to edit.
5. Hover over the pin, and then click the **Show more** icon.
6. Click **Edit**.
7. Update the pin title, image, or description.
8. Click **Submit**.

Deleting a pin from a space

Delete a pin from a space when the pin is no longer relevant. For example, in a Hiring space, you can delete the job profiles of candidates who have not passed their interviews.

©2025 Pegasystems Inc

Before you begin:

Ensure that you have owner or member access to the space:

- Create a space to become the owner. For more information, see Creating a space.
- The owner of the space sets you as the owner. For more information, see Updating the owner of a space.
- Join the space to become a member. For more information, see Joining a space.

NOTE: If you are a member of a space, you can only delete pins that you create.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space in which you want to delete a pin.
4. On the **Board** tab, search for the pin that you want to delete.
5. Hover over the pin, and then click the **Show more** icon.
6. Click **Delete**.
7. Click **Submit**.

©2025 Pegasystems Inc

Managing a space

Ensure that a space has correct information and relevant members by updating the details of the space and by adding and removing members. As an owner of a space, you can also approve or reject requests to join the space and set another member as the owner of the space.

The following Tasks can help you manage a space:

Updating details of a space

Update the details of a space to ensure that the information about the space is correct. For example, you can update the name of the space to be more specific and meaningful.

> **Before you begin:**
>
> Ensure that you have owner access to the space:
>
> - Create a space to become the owner. For more information, see Creating a space.
> - The owner of the space sets you as the owner. For more information, see Updating the owner of a space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space for which you want to update the details.

©2025 Pegasystems Inc

4. Click **Edit space**.
5. In the **Edit space** modal dialog box, perform any of the following actions:
   - Update the name, description, type, or image for the space.
   - Enable or disable task tracking.
   - Update the hierarchy by adding, removing, or updating the parent space.

     If you add a parent space, the current space becomes a subspace for the parent space. If you remove the parent space, the current space is no longer a subspace.
6. Click **Submit**.

> **TIP:** If the space is no longer required, delete the space by clicking **Actions** > **Delete**.

Managing members of a space

Ensure that a space engages relevant stakeholders by adding and removing members. You can also approve or reject requests to join the space.

> **Before you begin:**
>
> Ensure that you have owner access to the space:
>
> - Create a space to become the owner. For more information, see Creating a space.
> - The owner of the space sets you as the owner. For more information, see Updating the owner of a space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.

©2025 Pegasystems Inc

- To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
- To view only the spaces of which you are a member or an owner, click **Member of**.
- To view the spaces that are marked as promoted, click **Promoted**.

3. Click the space for which you want to manage the members.
4. Click **Add member**.
5. In the **Manage members** modal dialog box, perform any of the following actions:
   - To add a member to the space, in the drop-down list, press the Down arrow key to select the user that you want to add as a member, and then click the **Add member** icon.
   - To remove a member from the space, click the **Delete member** icon next to the name of the member.
   - To approve or reject member requests to join the space, click the **Approve member** or **Reject member** icon next to the name of the user.
6. Click **Submit**.

Updating the owner of a space

Update the owner of a space if you do not want or need to own the space, or can no longer own the space. For example, if you are moving to another project in your organization, appoint a new owner of the space.

> **Before you begin:**
>
> Ensure that you have owner access to the space:
>
> - Create a space to become the owner. For more information, see Creating a space.
> - Ensure that the space details are up-to-date because after you resign from being the space owner, you cannot edit any details of the space.

©2025 Pegasystems Inc

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space for which you want to update the owner.
4. On the **Activity** tab, click **Edit space**.
5. In the **Owner** section, click the **Change owner** icon next to the name of the current owner of the space.
6. Press the Down arrow key and select a member to set as the new owner.
7. Click **Submit**.

> **Result:**
>
> You no longer own the space, but you continue to be a member of the space.

Adding content to a space

Attach relevant correspondence and documentation to a space so that you concentrate the supporting information in the space. For example, you can add sales orders from your customers so that the orders are available to other users of your space.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
   - To view only the spaces of which you are a member or an owner, click **Member of**.
   - To view the spaces that are marked as promoted, click **Promoted**.
3. Click the space to which you want to add content.

©2025 Pegasystems Inc

4. In the upper-right corner, click the **Expand the utility pane** icon, and then click the **Related** tab.
5. In the **Files & documents** section, click the **Manage files and documents** icon, and then in the **Manage content** modal dialog box, select the type of content that you want to add to the space:
    - Create a new document by using the rich text editor.
    - Add an existing document to reuse content from another space.
    - Add a URL, for example, the address of a customer's website.
    - Upload a local document.
    - Select a file from an external repository. For more information, see Sourcing attachments from external storage.
6. Click **Submit**.

Managing content attached to a space

Enhance collaboration with other users of your application by providing additional information about the attached documents in Pulse, and by performing other actions on the attached files that help you control the content. For example, you can prepare a draft of a presentation that your team is to deliver and discuss the content in Pulse.

1. In the navigation pane, click **Spaces**.
2. Choose the spaces in your application to view.
    - To view the private and public spaces, and unlisted spaces of which you are a member, click **All**.
    - To view only the spaces of which you are a member or an owner, click **Member of**.
    - To view the spaces that are marked as promoted, click **Promoted**.
3. Select a space that contains a document that you want to discuss.
4. In the upper-right corner, click the **Expand the Utility pane** icon, and then click the **Related** tab.
5. In the **Files & documents** section, click the **Preview** link of the attached document.

©2025 Pegasystems Inc

6. Click the **Expand the utility pane** icon.
7. On the **Activity** tab, use the Pulse feed to discuss the document with other users. For more information, see Posting a message in Pulse, Replying to a message in Pulse, and Managing Pulse messages.

For example:

Collaborating on a file in a space by using Pulse

8. In the document header, perform additional actions on the document by completing any of the following actions:

Choices	Actions
Receive notifications for the document when you are not the owner	Click the **Follow** icon.
Modify the document	Click **Edit**.
Download the document	Click **Download the file**.
Add tags to the document	Click **Actions**, and then select **Manage tags**.

©2025 Pegasystems Inc

Choices	Actions
Modify your notification preferences for the document	Click Actions, and then select Manage Notifications.
Pin the document to a space or to the Recents list	Click Actions, and then select the relevant option.
Copy a link to the document	Click Actions, and then select Copy shareable link.
Delete the comment	Click Actions, and then select Delete.

Fine-tuning collaboration by creating subspaces

Improve collaboration on a topic by branching the parent space. In this way you can create multiple subspaces that contain relevant subtopic content for their members, grouped under their parent topic. For example, you can divide the Loans space into subspaces for different loan types, such as Car Loans or Mortgage Loans. In these subspaces, Case workers can access only the information for the loan types on which they are working.

> **Before you begin:**
>
> To create a subspace, you need to be a member of the parent space. For more information, see Joining a space and Creating a space.

You can add multiple subspaces to a parent space. However, a subspace can have only one parent space. You can also create subspaces within subspaces.

1. In the navigation pane, click Spaces.
2. Choose the spaces in your application to view.
   - To view the private and public spaces, and unlisted spaces of which you are a member, click All.
   - To view only the spaces of which you are a member or an owner, click Member of.

©2025 Pegasystems Inc

- To view the spaces that are marked as promoted, click **Promoted**.

3. Click the space within which you want to create a subspace.
4. Click **Actions** > **Add subspace**.
5. In the **Create a new subspace in** Parent space name modal dialog box, perform steps 4 through 9 in Creating a space.

For example:

A subspace within a parent space

6. **Optional:** To edit or adjust the subspace, perform any of the steps in Managing a space, Adding content to a space, and Managing content attached to a space.

Result:

You can find the newly added subspace on the **Subspaces** tab of the parent space and also on the Spaces landing page. The subspace is created with a Pulse interface for discussions, a link to the parent space above the title of the subspace, and sections that contain the subspace details.

©2025 Pegasystems Inc

What to do next:

Users can perform the same tasks with subspaces as they can with spaces, for example, create further subspaces, define visibility, and set access rights. A user can join a subspace and members can communicate with other members of the subspace. The owner can manage the members and the subspace details, such as updating or removing the parent space.

Creating and managing tasks in a space

Resolve tasks more effectively when collaborating with other users of your space by monitoring work on a Task board. Add tasks to your board so that all the space members are well-informed about the Assignments that they need to complete. You can notify all the members of the space by creating new tasks, updating the status of existing items, and adding checklists to tasks.

For example, a team of salespeople in a large company wants to prepare a presentation for their customers. Members of the team need a single tracking solution because they are located at different sites.

Before you begin:

Create a space and enable task tracking. For more information, see Creating a space.

1. In the navigation pane of the Portal that you use, click **Spaces**, and then choose the space that you want to update with tasks.
2. On the **Tasks** tab, in the **Open** column, click the **Create task** icon.
3. In the **Add task** modal dialog box, enter a name for the task.
4. Click **Create**.
5. **Optional:** To create more tasks, repeat steps 2 through 4.

©2025 Pegasystems Inc

For example:

You can create the following list of tasks for a presentation assignment:

- Collect additional information
- Conduct research for prognosis
- Create draft presentation
- Prepare graphics

6. **Optional:** To edit the task details, click the task, and then in the Task details modal dialog box, update the relevant items:

Choices	Actions
Change the task name	In the Task name field, provide a new name for the task.
Change the task status	In the Status field, select a new status for the task.
Change the owner of the task	In the Assignee field, select a new owner for the task.
Change the due date of the task	In the Due date field, select a new due date for the task.
Provide additional details about the task	In the Additional details field, enter relevant information about the task.
Enable quick filtering and checking the progress of tasks by adding task categories	a. In the **Category** section, in the Add category field, define the categories for the task by entering the name of the category or selecting the category from the list. **NOTE:** The list only contains the categories

©2025 Pegasystems Inc

Choices	Actions
	that you create in the space that you currently use. b. Click **Submit**.
Provide more detailed, smaller action items in the task	a. In the **Checklist** section, click the **Add checklist item** icon for every action item that you want to add. b. Click **Submit**.
Post a Pulse message about the task	In the Pulse section, in the **Start a conversation** field, enter your message. For more information, see Posting a message in Pulse.

For example:

Editing the category and checklist of a task

©2025 Pegasystems Inc

7. **Optional:** To add content to a task, hover over the task, and then click **Show more** > **Add content**:

Choices	Actions
Provide content in a rich text editor	a. In the **Manage content** modal dialog box, click **Rich text**. b. In the **Name** field, enter the name of the document, for example `Presentation agenda`. c. In the **Content** field, create the document. **For example:** Enter the presentation agenda: *Adding content to a Task in the rich text editor* d. Click **Submit**.
Upload a file	a. In the **Manage content** modal dialog box, click **Local file**, and then select a file that you want to upload. b. In the **Name** field, enter the name of the file, for example `Graphics - charts`. c. **Optional:** To provide additional information about the file, in the

©2025 Pegasystems Inc

Choices	Actions
	Description field, enter relevant details. d. Click Submit.
Pin a document	a. In the **Manage content** modal dialog box, click Add existing. b. In the autocomplete field, press the Down arrow key, and then select a document from your application, for example `Sales results file`. c. Click Add. d. Click Submit.
Add a URL	a. In the **Manage content** modal dialog box, click URL. b. In the Subject field, enter the name that you want to use for the URL, for example, if you add a URL to a site of a prospective client, enter the client's company name. c. In the URL field, enter the Internet address, for example `http://uplustelco.com`. d. Click Submit.

8. **Optional:** To effectively process your Case, on the Tasks tab, manage your tasks in one of the following ways:

Choices	Actions
Change the priority of a task	Drag the task up or down within a column.

©2025 Pegasystems Inc

Choices	Actions
Change the status of a task	Drag the task to one of the following columns: • Open • In-Progress • On-Hold • Completed **NOTE:** You can also change the task status in the task properties, which you configure in the **Task details** modal dialog box of a task, in the Status field. You can also mark a task as completed by clicking the Complete task button.
Filter tasks to display only tasks with a specific category	a. Click the Filter icon, and then select a category, for example `Collect data`. b. Click Apply.
Add content to a task	a. Hover over the task, and then click Show more > Add content. b. Perform any number of actions from step 7.
Edit task details	a. Click the task, and then update relevant information, for example, change the assignee or due date of the task. b. Click Submit.

©2025 Pegasystems Inc

Choices	Actions
Delete a task	a. Hover over the task, and then click Show more > Delete. b. Click OK.
Edit the checklist to inform other users about the progress of a task	a. Click the task, and then, in the **Checklist** section, select a checkbox next to an item that you want to mark as complete. b. Click Submit.
Post a Pulse message about the task	In the Pulse section, in the Start a conversation field, enter your message. For more information, see Posting a message in Pulse.

Result:

The Task board in your space now contains tasks that you process and clear to complete the assignment. The tasks are enriched with attached content to facilitate communication between space members. You can also track and manage the progress of a Case by updating the status of the tasks.

Granting Super Admin privileges to users

Provide users with Super Admin privileges to manage and edit any space, even if they are not members of the space. By authorizing certain users to manage any space in you application, you can manage spaces more effectively and have greater control over the content that space owners and members create.

Users with Super Admin privileges can perform the following actions in any space:

- Mark a space as promoted.

©2025 Pegasystems Inc

- Edit a space: change the space type, edit the name or description, and change the owner and hierarchy.
- Delete a space.
- Create subspaces.

1. Create an Access Group in the application in which you want to grant Super Admin privileges to a user.
   For more information, see Creating an Access Group.
2. Create an Access Role for promoted spaces.
   For more information, see Creating an Access Role by using the Rule form.

   > **TIP:** To save time, clone an existing Access Role.

3. Add the `PegaSocial-Group` class to the Access Roles.
4. Add the *pxSpaceAdmin* privilege with level 5 and give access level 5 to the following functions:
   - Read
   - Write
   - Delete
   - Run a report of Rules and instances

   Level 5 is the highest level.
5. Add the new Access Role to the new Access Group.
   For more information, see Adding a role to an Access Group.
6. Add the new Access Group to the users to whom you want to give Super Admin privileges, and then make the newly created Access Group the default Access Group.

Searching for Pulse messages in spaces

Access the information that you need more quickly by searching for Pulse messages within the context of a space. For greater accuracy, apply filters, such as keywords, and

©2025 Pegasystems Inc

message authors. For example, you can filter messages that include the phrase `Loan request`.

> **Before you begin:**
>
> Enable Search indexing and index Pulse instances. For more information, see Enabling the indexing of classes.

Search results include private and public posts, comments, and attachments that match the filters that you apply. If the result is a comment, Pulse displays the complete thread so that you can see the context of the comment. For attachments, the filters evaluate both file names and contents.

1. In the navigation pane, click **Spaces**.
2. Open the space for which you want to search the Pulse messages.
3. On the **Activity** tab, in the **Search** field, click the **Expand search option** icon.
4. In the **Filter posts** dialog box, define the filters that you want to apply.

   > **For example:**
   >
   > In the **Contains** line, enter `Loan request`.

5. Click **Search**.

> **Result:**
>
> The results display posts, comments, and attachments that match the applied filters.

Collaborating on shared content by using documents

By creating documents in an application, you ensure that all users of the application have sufficient information to work in a business process and you reduce context

©2025 Pegasystems Inc

switching by limiting the need to use external documents. You also save time by sharing the files across multiple Cases and spaces.

For example, an approver for a Car Loan Case can collaborate on an address verification document with other approvers. The approver can open the document from the Documents landing page or the Case itself, and then use Pulse to consult other approvers.

> **NOTE:** The feature is not available in Constellation applications. You can only configure an application to launch the feature in a section-based UI in a new browser tab. For more information, see Adding traditional landing pages to the navigation menu and Launching traditional pages from Constellation portals.

Documents are files that contain additional information about Cases and, unlike attachments, can be shared with all users in your application.

The Documents landing page in a Portal consolidates all the important content for your application in one place.

The following image shows the Documents landing page in User Portal with all the documents that are available to a user within the application:

©2025 Pegasystems Inc

The Documents landing page in User Portal

The following image shows the collaboration interface for a document in User Portal:

The Pulse interface for a sample document

Documents are always created within the context of your application, and are not limited to a Case, a space, or a Pulse conversation. However, you can reference the same document from multiple Cases and spaces in your application.

You create documents directly in your application by using the built-in rich text editor or by uploading a file. When you create a document by uploading a file, you can provide a name for the document and use the rich text editor to add a meaningful description. If needed, you can also update documents later.

With documents, you have the following options for collaborating on Cases:

- Use Pulse to discuss a document with other users of your application.
- Follow documents.
- Pin documents to spaces.
- Link documents to Cases.
- Add attachments to a document.

©2025 Pegasystems Inc

- Download a document.
- Generate a link to a document and share it with other users.

The following tasks can help you use documents to share information with other users:

- **Creating a document in an application**
- **Discussing and managing a document**

Creating a document in an application

Create a document to share or discuss information with other users in your application. For example, you can create a document and then use Pulse to discuss the document with your team.

> **Before you begin:**
>
> Ensure that you have the Documents page added to the main menu of the Portal in your application. For more information, see Organizing the main navigation for a standard work Portal.

The user who creates a document becomes the owner of the document.

1. Depending on the purpose of your document, create a document in one of the following places:

Choices	Actions
To share general information, for example, the latest quarterly results of your organization, create a document on the Documents landing page.	a. In the navigation pane of your portal, click **Documents**. b. Click **Create document**. c. Enter a name for the document. d. On a selected tab, create document content in the rich text editor or select a file from a local or external repository or system.

©2025 Pegasystems Inc

Choices	Actions
	The options depend on your settings. By default, you can create a new document by using the rich text editor or upload locally saved documents. **NOTE:** To enable users to download documents from external sources, configure external sourcing options in your application. For more information, see Sourcing attachments from external storage. *An example of document creation options on the tabs of the New Document dialog box*
To add information to a Case, for example, an invoice in a Sales Order Case, create a document from the Case.	a. Open the Case in which you want to create a document, and then click Expand the utility pane. b. In the **Files & documents** section, click the Manage files and documents icon, and then in the

©2025 Pegasystems Inc

Choices	Actions
	Manage content dialog box, on a selected tab, create document content in the rich text editor, or select a file from a local or external repository or system. The options depend on your settings. By default, you can create a new document by using the rich text editor or by uploading a locally saved document. **NOTE:** To enable users to download documents from external sources, configure external sourcing options in your application. For more information, see Sourcing attachments from external storage. c. Click **Submit**. **Result:** When you upload a file or use the rich text editor, the system creates a document in the background.

©2025 Pegasystems Inc

Choices	Actions
To add information to a space, for example, a job profile in the Hiring space, create a document from the space.	a. Open the space in which you want to create a document, b. On the Activity tab, click Expand the utility pane icon, and then click the Related tab. c. In the **Files & documents** section, click the Manage files and documents icon, and then in the **Manage content** dialog box, on a selected tab, create document content in the rich text editor, or select a file from a local or external repository or system. The options depend on your settings. By default, you can create a new document by using the rich text editor or by uploading a locally saved document. **NOTE:** To enable users to download documents from external sources, configure external sourcing options in your application. For more information, see Sourcing attachments from external storage. d. Click Submit.

©2025 Pegasystems Inc

Choices	Actions
	Result: When you upload a file or use the rich text editor, the system creates a document in the background.

2. **Optional:** To provide only relevant content for the users, grant access to the document to members of a space or Case by performing the following actions:
   a. In the **New Document** dialog box, in the **Available to** section, select Limited.
   b. In the Select type list, select Space or Case.
   c. In the Name field, enter the name of a space or Case.
3. Click Publish.

Result:

The document is added within the context of the application and is displayed on the Documents landing page.

Discussing and managing a document

Discuss a document by using Pulse to review additional information about a Case with other users in your application, for example, to analyze a faulty sales order.

1. In the navigation pane of your portal, click Documents.

   Result:

   The Documents landing page opens.

2. Click the document that you want to discuss.

©2025 Pegasystems Inc

3. On the **Activity** tab, use Pulse to discuss the document with other users.

   For more information about using Pulse, see Collaborating with users by using Pulse.

4. **Optional:** To modify the document by adding and managing additional elements, perform the following actions:

Choices	Actions
To receive notifications for the document when you are not the owner	In the header of the document, click the **Star** icon. **NOTE:** To unfollow the document, click the **Star** icon again.
To modify the following elements: • **Name** • **Attachments** • **Description** • **Visibility**	a. In the header of the document, click **Edit**, and then modify any of the available options. b. Click **Publish**.
To save an offline copy of the document when the document is an uploaded file	In the header of the document, click **Download the file**.
To perform any of the following actions: • **View document history** • **Add or manage tags** • **Modify your notification preferences for the document** • **Refresh the document page**	In the header of the document, click **Actions**, and then select the relevant option.

©2025 Pegasystems Inc

Choices	Actions
• **Copy a shareable link to the document** • **Pin the document to a space or the recent items list** • **Delete the document**	
To add and manage the following items: • **Attachments** • **Followers** • **Stakeholders** • **Tags**	Click Expand the Utility pane, and then, on the Related tab, edit the relevant section. *A sample document with selected attachments, stakeholders, followers, and tags*
To add and manage references to Cases, spaces, and other documents	Click Expand the Utility pane, and then, on the About tab, edit the **Recent references** section.

©2025 Pegasystems Inc