



Conversational Channels

17 October 2023



CONTENTS

Conversational channels.....	4
Pega Email Bot overview.....	5
Pega Email Bot features.....	7
Pega Intelligent Virtual Assistant overview.....	14
Pega Intelligent Virtual Assistant features.....	15
Conversational channel user roles.....	19
Text analysis concepts.....	21
Exploring text predictions and text analyzers.....	23
Implementing an email bot for an Email channel.....	25
Creating an Email channel.....	26
Defining Email channel behavior.....	29
Configuring intelligent email routing.....	44
Training the model for the Email channel.....	48
Triaging incoming emails.....	55
Viewing reports for the Email channel.....	72
Customizing the Email channel.....	76
Troubleshooting the email bot.....	95
Updating the email bot to the 8.7.1 version.....	99
Updating to the threading mechanism available in the 8.6 version.....	99
Updating the email bot to the 8.5 version.....	101
Creating an Email channel from an existing channel.....	110
Implementing an Intelligent Virtual Assistant for a conversational channel.....	112
Creating a Digital Messaging channel.....	113
Defining conversational channel behavior.....	172
Simulating a conversation and building a chatbot.....	199
Training the model for the IVA channel.....	205

Troubleshooting the conversational channel.....	213
Creating a Legacy Webchat channel.....	222
Creating an Alexa channel.....	261
Creating an IVA channel from an existing channel.....	266

Conversational channels

Conversational channels help organizations reach additional users of their enterprise applications. With conversational channels, users can seamlessly interact with a Pega Platform™ application to obtain help, request a service, and report or solve an issue by using Facebook Messenger, WhatsApp Messenger, Apple Messages for Business, SMS/MMS (Twilio), an embedded chat window, Amazon Alexa, or email.

A channel is the technology or an application with which a user interacts with your application. Pega Platform supports different types of channels. You add each channel separately to the system, and configure and extend each channel to fit the needs and requirements of your organization. For example, you configure an Email channel so that users can interact with your application by using an email bot. Users also interact with an application in a conversational way by using chatbots, the part of an application that runs natural language processing (NLP), decision-enabled analysis, case processing, and artificial intelligence.

Supported channels

You build Pega Intelligent Virtual Assistant™ (IVA) and Pega Email Bot™ as channels from the channel interface dashboard, to intelligently respond to users and run actions. A user can communicate with a Pega Platform application using one of the following integration components:

Email

Send email messages to interact with an email bot.

Digital Messaging

Send text messages with the integrated messaging platform using Apple Messages for Business, Facebook Messenger, MMS/SMS (Twilio), Web Messaging, WhatsApp Messenger, and a native Android or iOS app. For more information, see [Digital Messaging channel overview \(on page 116\)](#).

Amazon Alexa

Use voice commands to interact with an IVA for Alexa.

What's New

The following video describes new chatbot and email bot features that are available in Pega Platform version 8.7:

https://players.brightcove.net/1519050010001/default_default/index.html?videoId=6286653092001





Note: To learn more about new chatbot and email bot features in Pega Platform version 8.7, see [What's New in conversational channels 8.7](#).

[Pega Email Bot overview \(on page 5\)](#)

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

[Conversational channel user roles \(on page 19\)](#)

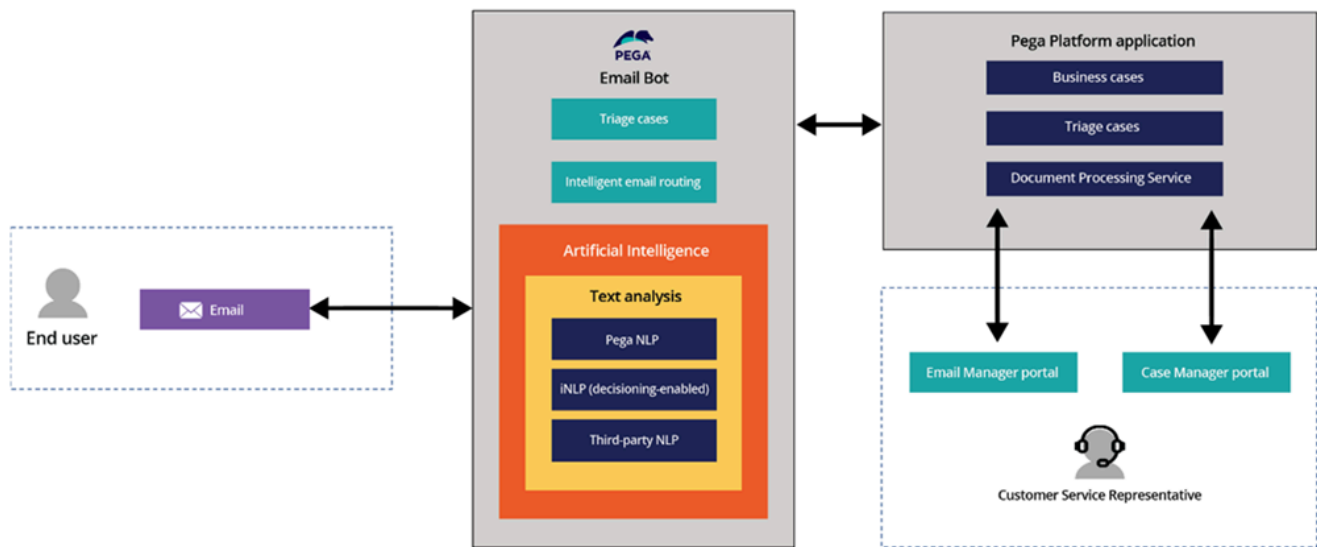
Pega Email Bot overview

Pega Email Bot™ is a bot system that intelligently interacts with your application to help users with their problems and speed up business processes. Interacting with an email bot through emails, users can more efficiently address their concerns and resolve problems, for example, by requesting more information and opening a business case in the system.

Different types of users work with the email bot for an application. A channel developer creates an Email channel that represents the email bot, defines its behavior, and trains data for the system. Customer service representatives (CSRs) and managers then triage incoming emails in the Email Manager or Customer Manager portal, to address the reported issue or customer request. A channel administrator troubleshoots and updates the email bot so that the system runs efficiently and smoothly. The end users are the customers that interact with the email bot in a production environment, by sending emails to report problems or to ask for more information.

Email bot components

An email bot that a channel developer builds and configures in a Pega Platform™ application consists of the following features and components:



Pega Email Bot components

Triage case

An internal object that the email bot, a Pega Platform application, and CSRs use to track user issues and requests from an email during a triage life cycle. The email bot automatically creates a triage case for each received email. CSRs or managers can then advance the triage case through its life cycle. For example, a CSR can reply to users by email, send Pulse messages, spin off a related business case, or resolve the triage case after the reported issue is adequately addressed in the system. For more information, see [Email triage process \(on page 57\)](#).

Intelligent email routing

A key feature of an email bot that allows the system to instantly and intelligently react to user requests and automatically take action based on text analysis of received emails. For example, an email bot converts the content of received emails into a new triage case, and then adds the email to the appropriate work queue, creates a top-level business case, and sends an automatic reply to the customer. To perform intelligent email routing, a channel developer sets up routing conditions for the system. For more information, see [Configuring intelligent email routing \(on page 44\)](#).

Text analysis

The capability of an email bot that examines the content of emails by using natural language processing (NLP), adaptive analytics, and artificial intelligence to seamlessly interact with users. The email bot can detect the general subject matter of the email (topic), text that falls into a general category (entities), sentiment, and language in the email by using text analysis. To perform text analysis, a data scientist configures a text prediction for the channel. A channel developer can also define text analyzers for the email bot, for example, Pega NLP.

For more information, see [Text analysis concepts \(on page 21\)](#) and [Defining Email channel behavior \(on page 29\)](#).

Channel behavior

The built-in artificial intelligence and text analysis capability of an email bot that ensures that the system responds correctly and promptly to users by email. Channel developers define the channel behavior of an email bot by adding a conversation to a case type, adding suggested cases and responses, and configuring text analyzer settings. Channel developers can then continue to improve the channel behavior by testing routing conditions used in intelligent email routing, training sample data, and rebuilding the text analytics model. For more information, see [Defining Email channel behavior \(on page 29\)](#) and [Training the model for the Email channel \(on page 48\)](#).

Email Manager and Case Manager portal

A portal in Pega Platform application that is available at run time to CSRs and managers, so that they can address issues reported by users quickly and more efficiently. CSRs use these portals to triage incoming emails from users that request information or report issues. For more information, see [Triage incoming emails \(on page 55\)](#).

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Conversational channel user roles \(on page 19\)](#)

[Conversational channels \(on page 4\)](#)

Pega Email Bot features

Pega Email Bot™ provides you with important features, depending on your version of Pega Platform™. Once you create, configure, and build an Email channel, users can send emails to your application to instantly ask for help, obtain more information, or report problems. The comparison matrices in the sections below provide a comprehensive view of feature support across different Pega Platform versions.

To learn more about email bots, see [Pega Email Bot overview \(on page 5\)](#). For relevant training materials, see the [Intelligent email](#) module on Pega Academy.

The following video describes new chatbot and email bot features that are available in Pega Platform version 8.7:

https://players.brightcove.net/1519050010001/default_default/index.html?videoId=6286653092001

Run time

The following table lists the available run-time features by Pega Platform version.



Run-time feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Show notification banners, and resolve or transfer a triage case to indicate that customer service representatives (CSRs) performed these actions.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Display the Get next email button in the Case Manager portal when you configure the Email channel.		✓							
Rename categories to topics.		✓	✓	✓	✓	✓	✓	✓	✓
Supports automatic mapping of entities to case properties.			✓	✓	✓	✓	✓	✓	✓
Check for duplicate cases.			✓	✓	✓	✓	✓	✓	✓
Use an improved user interface when replying and creating cases.			✓	✓	✓	✓	✓	✓	✓
Supports right-clicking when mapping entities to case properties.			✓	✓	✓	✓	✓	✓	✓
Show the detected entities from attachments on the View Analysis page.			✓	✓	✓	✓	✓	✓	✓
Link a triage case to a business case automatically.				✓	✓	✓	✓	✓	✓
Use the <i>Created by</i> and <i>Updated by</i> properties in a triage case and a business case for work that is automated by the Email channel.					✓	✓	✓	✓	✓
Display built-in reports in the Email Manager portal to visualize performance and level of automation.							✓	✓	✓

Design time

The following table lists the available design-time features by Pega Platform version.

Design-time feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Supports multiple email accounts in the Email channel.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Edit the <i>Text Analyzer</i> rule directly from the Email channel.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Display the account status verification and account status messages in the system.		✓	✓	✓	✓	✓	✓	✓	✓

Design-time feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Select the Record training data checkbox so that the system adds feedback on manually triaged cases to the text analytics model.		✓	✓	✓	✓	✓	✓	✓	✓
Supports the Save as option in the Email channel.			✓	✓	✓	✓	✓	✓	✓
Display the History tab in the Email channel.				✓	✓	✓	✓	✓	✓
Add a reference operator to the Email channel.					✓	✓	✓	✓	✓
Let users select multiple supported languages for the email bot.						✓	✓	✓	✓

Intelligent routing

The following table lists the available intelligent routing features by Pega Platform version.

Intelligent routing feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Supports automatic routing and case creation.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Supports routing that is based on the detected topic, sentiment, language, or a custom <i>When</i> rule.			✓	✓	✓	✓	✓	✓	✓
Supports automatic replies that are configured in routing rules.			✓	✓	✓	✓	✓	✓	✓
Show topics for all detected languages in the Email channel configuration.						✓	✓	✓	✓
Supports multiple languages during the creation of cases and sending of automatic replies.						✓	✓	✓	✓

Cases

The following table lists the available case features by Pega Platform version.

Case feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Perform mapping of entities to business case properties from email content.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Supports the configuration of suggested cases.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use a section that you can update on the Behavior tab for suggested cases.			✓	✓	✓	✓	✓	✓	✓
Map entities to case properties on the View analysis page.			✓	✓	✓	✓	✓	✓	✓



Case feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Use text analytics model feedback when manually sending suggested replies.			✓	✓	✓	✓	✓	✓	✓
Use language-specific configuration for suggested cases.						✓	✓	✓	✓
Handle responses to automatic email notifications triggered from a case.								✓	✓
Associate and disassociate service requests from a triage case.								✓	✓

Email replies

The following table lists the available email reply features by Pega Platform version.

Email reply feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Send a customer an email reply directly from a triage case.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Send feedback to a customer by sending suggested replies.		✓	✓	✓	✓	✓	✓	✓	✓
Reply to an email from a business case by using Pulse.			✓	✓	✓	✓	✓	✓	✓
Supports having multiple recipients in an email reply by using the TO/CC/BCC fields together with validations.				✓	✓	✓	✓	✓	✓
Display information about the email sender and the receivers in a triage case.				✓	✓	✓	✓	✓	✓
Send emails to users using the <i>send email</i> API.					✓	✓	✓	✓	✓
Supports language-specific configuration in suggested replies.						✓	✓	✓	✓
Reply to all of the email recipients by clicking Reply all .							✓	✓	✓
Forward emails to additional stakeholders from a triage case.								✓	✓
Outbound email templates for replies and acknowledgment emails.								✓	✓
Secured email threading logic.								✓	✓
Select email templates from the email compose window.								✓	✓
Save email drafts.									✓
Warning notification describing an issue with the delivery of emails.									✓

Document processing

The following table lists the available document processing features by Pega Platform version.

Document processing feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Use the Pega NLP text analyzer for analysis of text-based documents attached to an email.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Supports optical character recognition (OCR) in attached image-based files (by using Partner solutions and integrations).		✓	✓	✓	✓	✓	✓	✓	✓
Supports conditions during file attachment analysis.		✓	✓	✓	✓	✓	✓	✓	✓
Map entities detected in file attachments to case properties.		✓	✓	✓	✓	✓	✓	✓	✓

NLP analysis

The following table lists the available NLP analysis features by Pega Platform version.

NLP analysis feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Perform NLP analysis of content in an entire email thread.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Suggest cases that are based on NLP, and add a triage case to a business case along with file attachments.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Highlight detected entities in the body of an email.		✓	✓	✓	✓	✓	✓	✓	✓
Supports keyword-based entities.			✓	✓	✓	✓	✓	✓	✓
Supports the analysis of email subjects and signatures.				✓	✓	✓	✓	✓	✓
Supports named entity recognition, including multiple entities in the text analytics model.				✓	✓	✓	✓	✓	✓
Supports the NLP analysis of HTML content.				✓	✓	✓	✓	✓	✓
Supports the iNLP text analyzer and use outcomes from multiple text analyzers.					✓	✓	✓	✓	✓
Change the default analyzer in the advanced configuration and display the text analysis settings on the Behavior tab.					✓	✓	✓	✓	✓
Use language-specific configuration when editing topics.						✓	✓	✓	✓
Perform advanced text analyzer configuration in App Studio.							✓	✓	✓

Training models

The following table lists the available training model features by Pega Platform version.

Training models feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Use the Training data tab to perform inline editing of records.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Send feedback about training data to the text analytics model in the Email channel.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use bulk Mark reviewed and Delete actions on the Training data tab.		✓	✓	✓	✓	✓	✓	✓	✓
Update the text analytics model with one click.			✓	✓	✓	✓	✓	✓	✓
Edit and mark up the training data records on the Training data tab.			✓	✓	✓	✓	✓	✓	✓
Create, update, and remove entities on the Training data tab.			✓	✓	✓	✓	✓	✓	✓
Highlight entities.			✓	✓	✓	✓	✓	✓	✓
Create both entity and topic feedback for training records.			✓	✓	✓	✓	✓	✓	✓
Display warnings and errors after building the text analytics model.			✓	✓	✓	✓	✓	✓	✓
Click Save as to create a copy of out-of-the-box entities in the application ruleset.			✓	✓	✓	✓	✓	✓	✓
Export both reviewed and unreviewed records together with entity models.					✓	✓	✓	✓	✓
Import both reviewed and unreviewed records together with entity models.					✓	✓	✓	✓	✓
Send feedback to the text analytics model when removing an entity.					✓	✓	✓	✓	✓
Supports using different versions of the text analytics model.					✓	✓	✓	✓	✓
Supports multiple languages on the Training data tab.						✓	✓	✓	✓
Train the text analytics model based on email attachments.								✓	✓

Entity property mapping

The following table lists the available entity property mapping features by Pega Platform version.



Entity property mapping feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Use an entity-to-case property mapping definition at design time.			✓	✓	✓	✓	✓	✓	✓
Train entities when property mapping from a case, when the triage case data is manually mapped into a business case.				✓	✓	✓	✓	✓	✓
Train entities during the mapping of entities to case properties, including in HTML mode.					✓	✓	✓	✓	✓
Use a <i>When</i> rule to enable the collection of feedback for entity-to-case property mapping.						✓	✓	✓	✓
Use language-specific training during entity-to-case property mapping.						✓	✓	✓	✓

Email Manager portal

The following table lists the available Email Manager portal features by Pega Platform version.

Email Manager portal feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Assign triage cases to an operator based on their worklist.			✓	✓	✓	✓	✓	✓	✓
Refresh the worklist automatically to display new emails.				✓	✓	✓	✓	✓	✓
Refresh the user interface for the email list.				✓	✓	✓	✓	✓	✓
Use the Customize table option in the Email Manager portal.				✓	✓	✓	✓	✓	✓
Communicate internally using the Pulse tab and notifications.				✓	✓	✓	✓	✓	✓
Let operators access multiple work queues in the Email Manager portal.					✓	✓	✓	✓	✓
Use the Assign to self option in the work queue context.					✓	✓	✓	✓	✓
Show email sentiment pattern (positive, negative, neutral) for all received emails in a triage case.							✓	✓	✓
Use email threads for customer and other stakeholders for a triage case.								✓	✓

[Conversational channel user roles \(on page 19\)](#)

[Conversational channels \(on page 4\)](#)

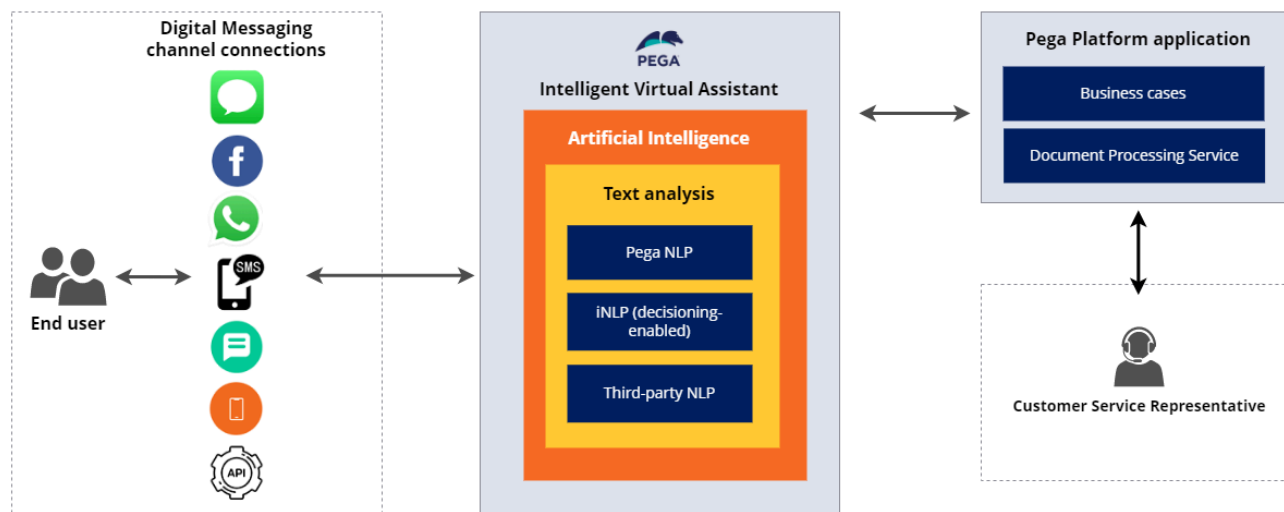
Pega Intelligent Virtual Assistant overview

Pega Intelligent Virtual Assistant™ (IVA) is a chatbot technology that interacts with an application by sending and receiving text messages from users, in a simple question and answer form. By interacting with an IVA through various social messaging platforms, users can resolve issues or address questions and other concerns. For example, to speed up business processes, users can request more details about a subject and open a case in the system.

Different types of users interact with an IVA for an application. A channel developer creates an IVA channel, configures its behavior, and trains data for the system. Customer service representatives (CSRs) and managers then use the Pega Customer Service portal to efficiently address the reported issues or other user requests. A channel administrator troubleshoots and updates the IVA so that the system runs correctly and smoothly. End users are customers that interact with an IVA in a production environment by sending text messages to report issues or to ask for help.

IVA components

An IVA that a channel developer builds and configures in a Pega Platform™ application consists of the following key features and components:



Pega Intelligent Virtual Assistant components

Social messaging platform

A way for users to interact with an IVA channel and a Pega Platform application to report an issue or resolve a problem using Digital Messaging Service. Users can choose Apple Messages for Business, Facebook Messenger, MMS/SMS (Twilio), Web Messaging, WhatsApp Messenger, and a native Android or iOS app as the messaging channel for an IVA for Digital Messaging. For more information, see [Digital Messaging channel overview \(on page 116\)](#) and [Digital Messaging Service overview \(on page 118\)](#).

Text analysis

A capability of an IVA that permits the system to examine the content of a chat text message by using natural language processing (NLP), adaptive analytics, and artificial intelligence to interact with a user in a natural, conversational manner. The IVA can detect the general subject matter of the email (topic), text that falls into a common category (entities), sentiment, and language in the message by using text analysis. To perform text analysis, a data scientist configures a text prediction or a channel developer defines text analyzers for the IVA, for example, Pega NLP. Next, a channel developer trains the data for the IVA in the preview console, so that the system knows how to correctly analyze and interpret content from chat text messages. For more information, see [Text analysis concepts \(on page 21\)](#) and [Adding a text analyzer for an IVA \(on page 188\)](#).

Channel behavior

The built-in artificial intelligence and text analysis capability of an IVA that ensures that the system responds correctly and promptly to users in a chat conversation. Channel developers define the channel behavior of an IVA by adding a conversation to a case type, adding case and responses commands to the system, and configuring text analyzer settings. Channel developers can then continue to improve the channel behavior by simulating a conversation in a preview console, training sample data, and rebuilding the text analytics model. For more information, see [Defining conversational channel behavior \(on page 172\)](#).

Preview console

A built-in panel in the IVA that channel developers can use to verify whether the chatbot works correctly by simulating a conversation before moving the system to a production environment. By making changes to the channel configuration and training data in a development environment, channel developers can rebuild the text analytics model to improve the text analysis and artificial intelligence of the IVA. For more information, see [Simulating a conversation and building a chatbot \(on page 199\)](#).

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

[Conversational channel user roles \(on page 19\)](#)

[Conversational channels \(on page 4\)](#)

Pega Intelligent Virtual Assistant features

Pega Intelligent Virtual Assistant™ (IVA) is a chatbot that provides you with several useful features, depending on your version of Pega Platform. After you create, configure, and build a chatbot, users can interact with your application by using a social messaging platform, such as Apple Messages for Business, Facebook, SMS/MMS, and WhatsApp, to instantly ask for help, obtain more information, or report problems. The comparison matrices in the sections below provide a comprehensive view of feature support across different Pega Platform versions.



To learn more about IVAs that work as chatbots, see [Pega Intelligent Virtual Assistant overview \(on page 14\)](#). For relevant training materials, see the [Digital Messaging channels](#) module on Pega Academy.

The following video describes new chatbot and email bot features that are available in Pega Platform version 8.7:

https://players.brightcove.net/1519050010001/default_default/index.html?videoId=6286653092001

Chatbot types

The following table lists available chatbots by Pega Platform version.

Chatbot type	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
IVA for Alexa	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVA for Legacy Webchat		✓	✓	✓	✓	✓	✓	✓	
IVA for Facebook (legacy channel)	✓	✓	✓	✓	✓	✓			
IVA for Facebook (Digital Messaging channel)						✓	✓	✓	✓
IVA for SMS/MMS - Twilio (Digital Messaging channel)						✓	✓	✓	✓
IVA for Apple Messages for Business (Digital Messaging channel)						✓	✓	✓	✓
IVA for Web Messaging (Digital Messaging channel)								✓	✓
IVA for WhatsApp (Digital Messaging channel)						✓	✓	✓	✓
IVA for Android app messaging (Digital Messaging channel)									✓
IVA for iOS app messaging (Digital Messaging channel)									✓
IVA for Client Channel API (Digital Messaging channel)									✓

Note:



- Due to changes by Twitter in the API and the licensing model, direct integration with Twitter is no longer supported in the Digital Messaging channel interface.
- In Pega Platform 8.7.1, IVA for Legacy Webchat is deprecated.
- IVA for Android and iOS app messaging requires Pega Mobile Messaging SDK.
- In Pega Platform 8.4 and 8.5, the IVA for Digital Messaging is called IVA for Unified Messaging.



- In Pega Platform 8.1 to 8.5, the IVA for Legacy Webchat is called IVA for Web Chatbot.
- In Pega Platform 7.4, the IVA for Legacy Webchat is called IVA for WebChat.

Run time

The following table lists available run-time features by Pega Platform version.

Run-time feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Rich user interface menu (IVA for Facebook and Legacy Webchat).		✓	✓	✓	✓	✓	✓	✓	✓
Date picker control (IVA for Legacy Webchat).				✓	✓	✓	✓	✓	✓
File attachments (IVA for Legacy Webchat).				✓	✓	✓	✓	✓	✓
Time and date picker control (IVA for Legacy Webchat).					✓	✓	✓	✓	✓
Simple forms (IVA for Legacy Webchat).					✓	✓	✓	✓	✓
Menu of commands displayable as simple menu buttons or text replies.							✓	✓	✓

Design time

The following table lists available design-time features by Pega Platform version.

Design-time feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Add new commands in the preview console.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Edit questions in the preview console.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Test the chatbot using the preview console.		✓	✓	✓	✓	✓	✓	✓	✓
Use an entity-to-case property mapping definition.			✓	✓	✓	✓	✓	✓	✓
Build a chatbot using the preview console.				✓	✓	✓	✓	✓	✓
Create a single channel to expose on multiple messaging platforms (Digital Messaging).						✓	✓	✓	✓
Create a stand-alone conversation process.								✓	✓
Configure conversation smart shapes in a conversation process: Ask a question, Send a message, and Conversation flow.								✓	✓

Design-time feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Let users select multiple languages for the Digital Messaging chatbot.									✓

NLP analysis

The following table lists available NLP analysis features by Pega Platform version.

NLP analysis feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Perform NLP analysis of each chat message from users.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Suggest case commands that are based on NLP in order to start a business case.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Highlight detected entities in the chat message in the preview console.		✓	✓	✓	✓	✓	✓	✓	✓
Supports decisioning capabilities in chat conversations.		✓	✓	✓	✓	✓	✓	✓	✓
Supports keyword-based entities.			✓	✓	✓	✓	✓	✓	✓
Supports named entity recognition, including multiple entities in the text analytics model.				✓	✓	✓	✓	✓	✓
Supports the iNLP text analyzer and uses outcomes from multiple text analyzers.					✓	✓	✓	✓	✓
Change the default analyzer in the advanced configuration and display the text analysis settings on the Behavior tab.					✓	✓	✓	✓	✓
Configure the chatbot to detect small talk separately.						✓	✓	✓	✓

Training models

The following table lists available training model features by Pega Platform version.

Training models feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Use the Training data tab to perform inline editing of records.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Send feedback about training data to the text analytics model in the IVA channel.		✓	✓	✓	✓	✓	✓	✓	✓
Use bulk Mark reviewed and Delete actions on the Training data tab.		✓	✓	✓	✓	✓	✓	✓	✓

Training models feature	7.3	7.4	8.1	8.2	8.3	8.4	8.5	8.6	8.7
Update the text analytics model with one click.			✓	✓	✓	✓	✓	✓	✓
Edit and mark up the training data records on the Training data tab.			✓	✓	✓	✓	✓	✓	✓
Create, update, and remove entities on the Training data tab.			✓	✓	✓	✓	✓	✓	✓
Highlight entities in the preview console.			✓	✓	✓	✓	✓	✓	✓
Create both entity and topic feedback records for text analytics models.			✓	✓	✓	✓	✓	✓	✓
Click Save as to create a copy of out-of-the-box entities in the application ruleset.			✓	✓	✓	✓	✓	✓	✓
Export both reviewed and unreviewed records together with entity models.					✓	✓	✓	✓	✓
Import both reviewed and unreviewed records together with entity models.					✓	✓	✓	✓	✓
Send feedback to the text analytics model when removing an entity.					✓	✓	✓	✓	✓
Text analytics model versioning.					✓	✓	✓	✓	✓
Supports multiple languages on the Training data tab for the Digital Messaging channel.									✓

[Conversational channel user roles \(on page 19\)](#)

[Conversational channels \(on page 4\)](#)

Conversational channel user roles

Pega Platform users with different roles create, train, and build Pega Intelligent Virtual Assistant™ (IVA) and Pega Email Bot™, as well as triage cases. For example, a channel developer creates an Email channel and builds its behavior, by training the text analytics model to better detect the correct topics, language, sentiment, and entities. Once the email bot is in a production environment, a customer service representative (CSR) then triages the cases so that customer emails are promptly and correctly addressed.

Supported roles

The following are the main user roles for creating, defining, and managing IVAs and email bots in Pega Platform:



Channel developer

Channel developers are in charge of developing the email bots and IVAs that interact with an application, before they are moved to a production environment. Channel developers typically perform the following tasks:

- [Learn text analysis \(on page 21\)](#) (prerequisite)
- [Configure an Email channel \(on page 26\)](#)
- [Create an Alexa channel \(on page 261\)](#)
- [Create a Digital Messaging channel \(on page 113\)](#) (for Apple Messages for Business, Facebook, MMS/SMS (Twilio), and WhatsApp social messaging channels).
- [Define Email channel behavior \(on page 29\)](#)
- [Define conversational channel behavior \(on page 172\)](#)
- [Train data for an Email channel \(on page 48\)](#)
- [Train data for a conversational channel \(on page 205\)](#)

Customer service representative (CSR)

CSRs interact with the application run time environment for email bots, and also directly with customers once the email bot is moved to a production environment. They use the Case Manager portal or the Email Manager portal for this purpose, to more efficiently address customer emails. Amongst other things, CSRs perform the following task:

- [Triage incoming emails \(on page 55\)](#)

Channel administrator

Channel administrators manage the IVAs and email bots for a Pega Platform application, once the IVA or email bot is in a production environment. An administrator works together with the channel developers to keep the system secure, operable, and error free. Administrators perform the following tasks:

- [Troubleshoot an email bot \(on page 95\)](#)
- [Update an email bot \(on page 101\)](#)
- [Troubleshoot an IVA \(on page 213\)](#)
- [Update an IVA](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

[Conversational channels \(on page 4\)](#)

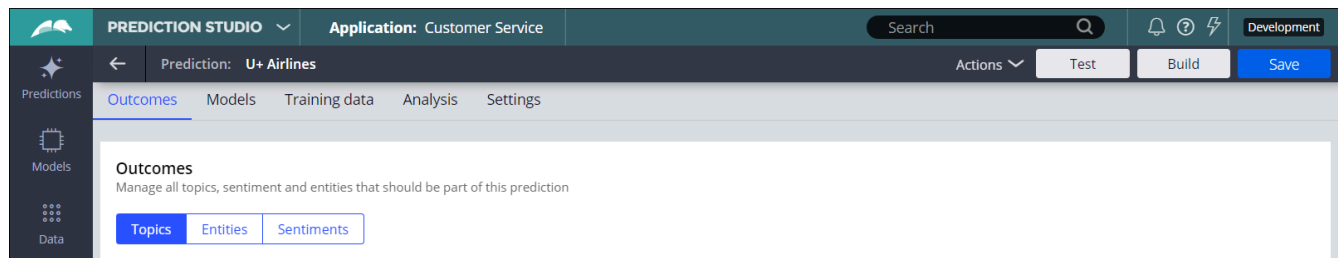
Text analysis concepts

Text analysis is an important aspect of conversational channels that enables a Pega Platform™ application to intelligently and seamlessly interact with a user in a natural conversational manner. Text predictions and text analyzers examine user input by using natural language processing (NLP), predictive and adaptive analytics, and artificial intelligence to find the best matching response.

Text predictions vs text analyzers

To use text analysis in conversational channels, configure a text prediction or one or more text analyzers for an Pega Intelligent Virtual Assistant™ (IVA) or Pega Email Bot™ in your application.

Text predictions are a new type of prediction in Prediction Studio that provides efficient configuration and monitoring of text analytics for your channels. The system automatically creates a text prediction for every channel that you add in your application. Every text prediction is associated with a text analyzer as its basis. You can open the text analyzer from the Behavior tab in your channel or from the Actions menu in the text prediction.



Text prediction

Text predictions are targeted to replace text analyzers that were used in previous versions of Pega Platform for analyzing messages in conversational channels. For more information about text predictions, see the following articles:

- [Text predictions simplify the configuration of text analytics for conversational channels](#)
- [Predict customer needs and behaviors through text analytics in your conversational channels](#)
- [Analyzing messages with text predictions \(on page \)](#)

Text analyzers are still available in Pega Platform and you can continue to use them. However, to get the best results out of your text analysis and to support more efficient updates, use text predictions.

Text analysis features

Text predictions and text analyzers detect the following information categories:

Topic



The general subject, an intent of an email, text message, or a voice command. An IVA or an email bot links all suggested cases and suggested responses to topics. For example, an email bot can determine that the topic of an email relates to a car insurance, and then open a car insurance business case.

Entity

The text contains proper nouns that fall into a common category, for example, a person, location, date, organization, or ZIP code. You can configure the IVA or the email bot to automatically assign the entities that they detect to properties of a new business case.

Sentiment

The opinion that a user expresses in an email, a chat text message, or a voice command: positive, neutral, or negative. An email bot can detect a negative sentiment of a user email, and then escalate the issue by automatically forwarding that information to a customer service representative.

Language

The language of an email, a chat text message, or a voice command. An email bot can detect the language of a user email, perform text analysis in that language by using NLP, and then automatically send a reply in this language to the user.

Text analysis in an email bot

Each text prediction or text analyzer configured for an email bot supports advanced text analysis of email header, body, subject, and attachments, including image files. To perform text analysis of image-based file attachments, you use the Pega optical character recognition (OCR) component that you install from Pega Marketplace on premises.

With text analysis and intelligent email routing, an email bot interprets an email and determines how to correctly respond to a user. This functionality also improves the triage process by creating a correct business case with the help of the email topic.

To refine the text analysis capability for your email bot, you can define multiple models in a text prediction to serve the same or different purposes.

Text analysis in an IVA

A text prediction or text analyzer that you configure for an IVA provides advanced text analysis of user input, including text voice commands. The text prediction helps the system determine the best matching response by using NLP, adaptive analytics, and artificial intelligence.

[Pega Email Bot overview \(on page 5\)](#)

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

[Conversational channels \(on page 4\)](#)

Exploring text predictions and text analyzers

Text predictions and text analyzers for Pega Email Bot™ and Pega Intelligent Virtual Assistant™ (IVA) process user input and help the system find the best matching response by using natural language processing (NLP) and predictive and adaptive analytics. You can configure text predictions and text analyzers to detect topics, entities, sentiment, and language in an email, chat text message, or a voice command.

Text analyzer types

Your Pega Platform application supports text predictions and different types of text analyzers that you can add and configure on the Behavior tab in your email bot or IVA channel:

Exact match

The default simple text analyzer that exactly matches user input to a response.

Pega NLP

An advanced text analyzer that determines the best approximate match by using advanced natural language processing (NLP) and artificial intelligence.

Text prediction / iNLP

The default text prediction for the channel is associated with the iNLP text analyzer, as shown in the following figure. The text prediction/iNLP text analyzer detects topics, entities, sentiments, and language in a message by using NLP, predictive and adaptive analytics, and artificial intelligence.



Tip: Text predictions are targeted to replace text analyzers. As a best practice, configure text analytics for your conversational channels by using text predictions instead of text analyzers. For more information, see [Text analysis concepts \(on page 21\)](#).

APP STUDIO Application : Customer Service

Legacy Webchat interface: U+ Airlines

Configuration Behavior Connection Training data Mashup

Text Analyzer ☐ Use advanced configuration

Edit topics

☐ Enable model training

☒ Enable small talk

[Open Text Prediction](#) New

While Text Prediction is targeted to replace Text Analyzer you can still access it.

[Open Text Analyzer in Dev Studio](#)

APP STUDIO Application : Customer Service

Legacy Webchat interface: U+ Airlines

Configuration Behavior Connection Training data Mashup

Text Analyzer ☒ Use advanced configuration

Text analyzers that process user's input and look for matching responses.

Type	Apply in context of a case	Apply outside of case context
1 Exact match	✓	✓
2 iNLP	✓	✓

[+ Add text analyzer](#)

Text analyzer configuration

Text Analyzer type: iNLP

Title	Context of a case	Outside of a case context
Entities	✓	✓
Topics	✓	✓

☐ Enable model training

☒ Enable small talk

[Open Text Prediction](#) New

While Text Prediction is targeted to replace Text Analyzer you can still access it.

[Open Text Analyzer in Dev Studio](#)

Text prediction and text analyzers in a channel

Text analytics model

Text analytics models provide algorithms, NLP, adaptive analysis, and artificial intelligence. To ensure that the text analyzer correctly detects a topic, entity, sentiment, or language, you first train the email bot or the IVA using the data, and then apply the changes to the text analytics model. For more information, see [Training the model for the Email channel \(on page 48\)](#) and [Training the model for the IVA channel \(on page 205\)](#).

[Text analysis concepts \(on page 21\)](#)

[Pega Email Bot overview \(on page 5\)](#)

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

[Conversational channels \(on page 4\)](#)

Implementing an email bot for an Email channel

Give users the ability to seamlessly interact with an application by email. When you implement an Email channel, your Pega Email Bot™ intelligently responds to user email requests. For example, the email bot opens a related business case about a reported issue and forwards the triage case to another operator by using intelligent email routing. A triage case stores information from a received email in the system.

In an email bot, you can set up intelligent routing of emails by defining routing and automation conditions based on rules, email details, or natural language processing (NLP), such as entity extraction, category detection, and sentiment analysis. With intelligent routing, the system automatically routes incoming email content to an operator or to a work queue as an email triage, or creates a case. When you set up an Email channel, Pega Platform™ automatically configures email instantiation for the selected case.

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

Note:



- If you are experiencing issues with the Theme Cosmos design of your Email Manager portal, download the [Email Manager Cosmos component](#) from Pega Marketplace.
- The process of creating a Pega Platform application, including the design and development of an Email channel for an email bot, requires open rulesets. However, after moving your email bot to a production environment, you can lock the rulesets. Only when you need to build models further or change the email bot channel configuration in a production environment, must you then unlock the rulesets.

1. Build and configure an Email channel for an application:

- a. Set up an email bot by creating an Email channel, adding email accounts for email handling, and selecting a reference operator.
For more information, see [Creating an Email channel \(on page 26\)](#).
- b. Define Email channel behavior by adding suggested cases, defining suggested responses, and configuring email analysis.
For more information, see [Defining Email channel behavior \(on page 29\)](#).

- c. Set up intelligent routing of emails by defining routing conditions based on email header content, text entity, topic, sentiment analysis, or a when rule.
With intelligent routing, the system automatically routes email content from users to an operator or to a work queue as an email triage case, or creates a top-level case. For more information, see [Configuring intelligent email routing \(on page 44\)](#).
2. Use the Email channel to record and train the data for the text analytics model, triage emails, and display reports:
 - a. Enable the recording of training data, create training data, and then apply changes to the text analytics model for the email bot.
For more information, see [Training the model for the Email channel \(on page 48\)](#).
 - b. Triage incoming emails for the email bot.
You can reply to users, forward emails to additional stakeholders, send Pulse messages, spin off business cases, detect duplicate triage cases, or resolve triage cases. For more information, see [Triaging incoming emails \(on page 55\)](#).
 - c. Display built-in reports about the number of received emails, triaged cases, and created business cases.
For more information, see [Viewing reports for the Email channel \(on page 72\)](#).
3. Manage the Email channel by troubleshooting, updating, and further customizing the system:
 - a. Perform additional tasks to customize the email bot.
For more information, see [Customizing the Email channel \(on page 76\)](#).
 - b. Address issues in the email bot by verifying the email integration settings and fixing the routing conditions that you define for intelligent routing.
For more information, see [Troubleshooting the email bot \(on page 95\)](#).
 - c. Perform maintenance tasks and update the email bot to the 8.5 version, to improve performance and make the system more reliable.
For more information, see [Updating the email bot to the 8.5 version \(on page 101\)](#).

[Pega Email Bot overview \(on page 5\)](#)

[Conversational channels \(on page 4\)](#)

Creating an Email channel

Ensure that your Email channel has integrations with necessary email accounts, and is properly configured for your system, for example, to support multiple languages and display your organization's unique branding in outbound emails. As a result, you ensure that users have an additional way to communicate



with your application to ask for help or solve an issue. For example, by creating an email bot for a travel agency you can help users book a flight by interacting with your application just by using email.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Create new channel interface** section, click **Email**.
4. In the **Email channel name** field in the **Details** section, enter a name for your Email channel.
5. **Optional:** To explain the purpose of your email bot, in the **Description** field, enter a short description for the email bot.
6. In the **Email handling** section, click **Add email account**, and then select an email account. You set up at least one email account for the Email channel so that users know which email address to send their requests to the email bot.
7. **Optional:** To verify that you can use the email account that you selected, click **Verify**. For more information about troubleshooting email integration settings for the Email channel, see [Troubleshooting email accounts in the Email channel \(on page 96\)](#).

If the email account verification fails, you can view information about the reason why the process failed by clicking a link.

8. If you want to define additional email accounts for the email bot, repeat steps [6 \(on page 27\)](#) and [7 \(on page 27\)](#).
9. **Optional:** To embed case identifiers in the email subject, in the **Additional settings** section, select the **Display case ID in subject** checkbox.
By default, the email bot uses a secure threading mechanism by embedding an encrypted security code in the email body to track emails replies from customers and other stakeholders in the correct email thread for a triage case. However, if your organization requires that emails display case identifiers in the email subject, you can enable this option. For more information, see [Email triage process \(on page 57\)](#).
10. **Optional:** To enhance how information is presented in emails and triage cases and to enable rich HTML text for the email bot, in the **Additional settings** section, select the **Allow rich text (html) in emails** checkbox.
Enabling this option enhances how important information in the email is presented in the system, for example, in triage cases displaying conversations details from users.
11. In the **Additional settings** section, in the **Reference operator** list, select a reference operator for your email bot.
You select a reference operator so that the system automatically assigns the email triage case to a unique Pega Platform™ operator when an exception occurs. A reference operator is a Pega Platform user on behalf of whom service mails are created in the system.
12. In the **Languages** list, select the languages in which the email bot interacts with users.

You select the languages in which the email bot processes emails so that the system can perform text analysis and provide automatic responses in the correct language. By supporting multiple languages in an email bot, you increase the availability of your application to users who speak different languages. You configure reply email templates for each language separately.



Note: By editing the `pyGetDefaultLanguage` data transform rule for the `Data-Channel` class, you can select a default language other than English to use when creating an Email channel. For more information, see [Overriding the default language for an Email channel \(on page 93\)](#).

13. In the **Outbound email template** list, select a template for the system to use for outbound emails. You select an outbound email template so that email interactions with customers and other stakeholders have a consistent look and feel that represents the unique brand for your organization. The system provides you with three built-in templates. The Cobalt template uses a solid blue header with a light-colored logo and the system displays email threads in separate bubbles with the same blue top border. The Classic template uses a darker color logo on a gray background and the system displays the email threads in separate bubbles with a white background. The Clear template does not use any visible styles.



Note: You can also create custom outbound email templates that you later select from the **Outbound email template** list. For more information, see [Creating outbound email templates \(on page 85\)](#).

14. **Optional:** To fit your business needs, modify the selected outbound email template:

- a. Click **Edit template** next to the **Outbound email template** list.

The system displays the outbound email template rule in Dev Studio.

- b. Modify how the template looks by changing the email background color, logo, header, body, and footer information.
- c. Save the outbound email template rule that belongs to a locked ruleset to an isolated sandbox so that you can test your changes, by clicking **Private edit**.
For more information, see [Performing a private edit \(on page 93\)](#).

- d. Save the outbound email template rule to a ruleset by clicking **Save as**.
- e. Click **Back to Pega App Studio**.

15. Click **Save**.

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Defining Email channel behavior

Ensure that Pega Email Bot™ responds to emails in an automated and intelligent way by precisely defining how the system behaves. As a result, you make the job of customer service representatives (CSRs) easier and increase the efficiency and responsiveness of your email bot for your application. For example, CSRs can save time responding to customer email requests when the email bot correctly detects the email subject, entities and language, and as a result provides meaningful replies and suggests relevant business cases to spin-off.

1. Provide the ability to start a top-level case based on emails that are received from users.
For more information, see [Adding suggested cases for an Email channel \(on page 30\)](#).
2. Provide meaningful automatic responses to emails.
For more information, see [Defining suggested responses for an Email channel \(on page 33\)](#).
3. Define how Pega Email Bot analyzes emails from users so that the email bot takes advantage of natural language processing (NLP) and adaptive analytics:
 - a. Configure text analyzers for the email bot so that the system users natural language processing (NLP) and adaptive analytics text analysis of the email.
For more information, see [Adding a text analyzer for an email bot \(on page 36\)](#).
 - b. Define topics, the general subject, the intent of email that is detected by the email bot using text analysis.
For more information, see [Defining topics for text analysis for an email bot \(on page 39\)](#).
 - c. **Optional:** To let the email bot analyze text contained in image-based email attachments, configure and use the optical character recognition (OCR) component.
For more information, see [Pega OCR on Pega Marketplace](#).
 - d. **Optional:** To enable the automatic detection of a language for text analysis, configure a dedicated text analyzer.
For more information, see [Enabling automatic language detection for text analysis \(on page 43\)](#).

When the email bot correctly detects the subject matter, sentiment, entities and the language of an email, you improve the automatic actions that the system can then perform on emails. The system sets up the iNLP text analyzer by default, which uses natural language processing (NLP) and adaptive analytics text analysis to detect topics and entities in emails.



Note: For more information about NLP and machine learning in email bots, see [Using email bots for natural language processing and machine learning](#).

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Adding suggested cases for an Email channel

Specify suggested business cases for your Pega Email Bot™ to provide meaningful automatic responses to emails. Based on text analysis of a received email, the email bot uses the suggested case as the detected subject matter, to match a response to the email. As a result, customer service representatives (CSRs) can react to user inquiries more quickly and efficiently, improving overall user satisfaction. For example, if you define a bank loan business case in your application and later add it as a suggested case for the email bot, when the system detects that an email request from a customer is related to a bank loan request, the system then suggests a bank loan case to spin off to the CSR that triages the email.

When you triage an email for an interaction case, you can spin off a regular case only if that regular case type is in the suggested cases list. For suggested cases that you define, the email bot can also perform automatic entity extraction of email contents, for example, city names, dates, or addresses, to case properties. You can also use suggested cases during intelligent email routing.

To learn about creating a case from email by using relevant training materials and a virtual Pega instance, see the [Creating a case from an email](#) challenge on Pega Academy.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. In the **Suggested cases** section, click **Add suggested case**.
6. In the **Case type** list, click the case type that you want to add.
7. In the **Suggested cases** section, select a method for modifying a suggested case:

- To create a new suggested case, click **Add suggested case**, and then, in the **Add suggested case** window in the **Choose case type** list, select a case type.
- To edit an existing suggested case, click the **Switch to edit mode** icon for the case type that you want to update.

For more information, see [Setting up text analysis for a suggested case \(on page 31\)](#).

8. If you configure multiple languages for the email bot, in the **Add suggested case** window, click a language.

The supported languages are displayed on a vertical tab. When you add a new case or edit an existing case, you switch between the languages by clicking the name of a language.

9. Enter the items that trigger an action, and then separate them with commas:
 - a. In the **Approximate match** field, enter words and short phrases to use as an approximate match for the suggested case that triggers the action.
 - b. In the **Must match** field, enter words and short phrases that trigger an action for the suggested case when they occur in the email.
 - c. In the **Never match** field, enter words and short phrases that never trigger an action for the suggested case.
10. **Optional:** To configure additional languages for the suggested case, repeat steps [8 \(on page 31\)](#) through [9 \(on page 31\)](#).
11. Click **Submit**.
12. **Optional:** To add more suggested cases, repeat steps [5 \(on page 30\)](#) through [11 \(on page 31\)](#).

[Defining Email channel behavior \(on page 29\)](#)

Setting up text analysis for a suggested case

Set up text analysis for a suggested case in an Email channel so that your Pega Email Bot associates a topic with the case and triggers an action as a response to a received email. With text analysis, your email bot provides meaningful responses to incoming email. For example, when the email bot detects the intent of an email that relates to a request for a car insurance quote, the system suggests a car insurance case.

If you configure multiple languages for the email bot, you set up text analysis for each language separately. This configuration ensures that the system performs email text analysis in the correct language.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your Email channel.
4. In the Email channel, click the **Behavior** tab.
5. In the **Suggested cases** section, select a method for modifying a suggested case:

- To create a new suggested case, click **Add suggested case**, and then, in the **Add suggested case** window in the **Choose case type** list, select a case type.
 - To edit an existing suggested case, click the **Switch to edit mode** icon for the case type that you want to update.
6. If you configure multiple languages for the email bot, in the **Add suggested case** window, click a language.
The supported languages are displayed on a vertical tab. When you add a new case or edit an existing case, you switch between the languages by clicking the name of a language.
 7. Enter the items that trigger an action, and then separate them with commas:
 - a. In the **Approximate match** field, enter words and short phrases to use as an approximate match for the suggested case that triggers the action.
 - b. In the **Must match** field, enter words and short phrases that trigger an action for the suggested case when they occur in the email.
 - c. In the **Never match** field, enter words and short phrases that never trigger an action for the suggested case.
 8. **Optional:** To configure additional languages for the suggested case, repeat steps 6 (on page 32) through 7 (on page 32).
 9. Click **Submit**.

At run time, the suggested case action starts when an email contains the items that you defined.

[Creating an Email channel \(on page 26\)](#)

[Defining Email channel behavior \(on page 29\)](#)

Setting up entity property mapping

To improve the responses that your Pega Email Bot™ provides to emails, you can optionally add entity extraction information for a suggested case. As a result, when a case type is associated with an email that you receive, your email bot extracts entities from email content, and automatically assigns them to a case type property in a Pega Platform application. For example, for the *Car Insurance* case, you can map the *CarModel* and *CarMake* entities to the `.CarModel` and `.CarMake` case properties. When these entities are detected in email, their values are automatically assigned to the case properties.

Entities contain proper nouns that fall into a commonly understood category, for example, a person, a location, a date, an organization, or a ZIP code.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.



3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. In the **Suggested cases** section, click the **Switch to edit mode** icon for a case type.
6. In the **Add suggested case** window, click the **Entities extraction** tab.
7. Add an entity to a case property mapping by clicking **Add mapping**.
8. In the **Entity** list, click an entity.
9. In the **Case property** list, click a case property that you want to map to the entity.
10. **Optional:** To add another entity to case property mapping, repeat steps [7 \(on page 33\)](#) through [9 \(on page 33\)](#).
11. Click **Submit**.

[Defining Email channel behavior \(on page 29\)](#)

Defining suggested responses for an Email channel

Set up suggested responses for Pega Email Bot™ so that customer service representatives (CSRs) can immediately select the most relevant reply to an email when they triage emails for an interaction case. A suggested response provides instant feedback to the customer in a clear and organized way, saving the CSRs time in responding to the initial inquiry. For example, when users request a bank loan by email, the email bot suggests a predefined email reply that the CSR can immediately send to users, asking for their address and financial history information.

When you define responses, you can create an email response template or use an existing template. If you configure multiple languages for the email bot, you set up text analysis for each language separately. This configuration ensures that the system performs email text analysis in the correct language.

For your convenience, the following default templates are available: apology, clarification, and missing information. To create additional default templates, define a Rule-Obj-Corr rule for the Work-Channel-Triage class.

Note: When you create a new suggested reply template in the Email channel, the system creates the template as a correspondence rule. You must create the template in an open rule set for your application by performing the steps in the Email channel described below.

To learn more about this feature by using relevant training materials and a virtual Pega instance, see the [Suggesting a reply based on email content](#) challenge on Pega Academy.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your Email channel.
4. In the Email channel, click the **Behavior** tab.
5. In the **Suggested replies** section, select a method for modifying a suggested reply:
 - To create a new suggested reply, click **Add suggested reply**.
 - To edit an existing suggested reply, in the list of replies, click the **Switch to edit mode** icon for the reply that you want to update.
6. In the **Add suggested reply** window, set up a predefined email response:

Choices	Actions
Use an existing template	a. In the Email template list, click a predefined email response.
Create a template	a. Click Create new . b. In the Name field, enter a label for the email response template. c. If you configure multiple languages for the email bot, in the Email template window, click a language. d. In the text area, enter the default content for the email response template. e. Optional: To configure additional languages for the suggested response, repeat steps 6.c (on page 34) through 6.d (on page 34) . f. Click Submit .
Edit a template	a. In the Email template list, select an email template. b. Click Edit . c. If you configure multiple languages for the email bot, in the Email template window, click a language. d. In the text area of the Email template window, make your edits. e. To configure additional languages for the suggested response, repeat steps

Choices	Actions
	6.c (on page 34) through 6.d (on page 34). f. Click Submit.

- In the **Associate topics** list, select the topics that you want to associate with the suggested response.

Associate each suggested email response with at least one topic so that only relevant responses appear during the triage process. You can associate more than one topic with a suggested response.

For more information, see [Defining topics for text analysis for an email bot \(on page 39\)](#).

- Click **Submit**.

[Creating an Email channel \(on page 26\)](#)

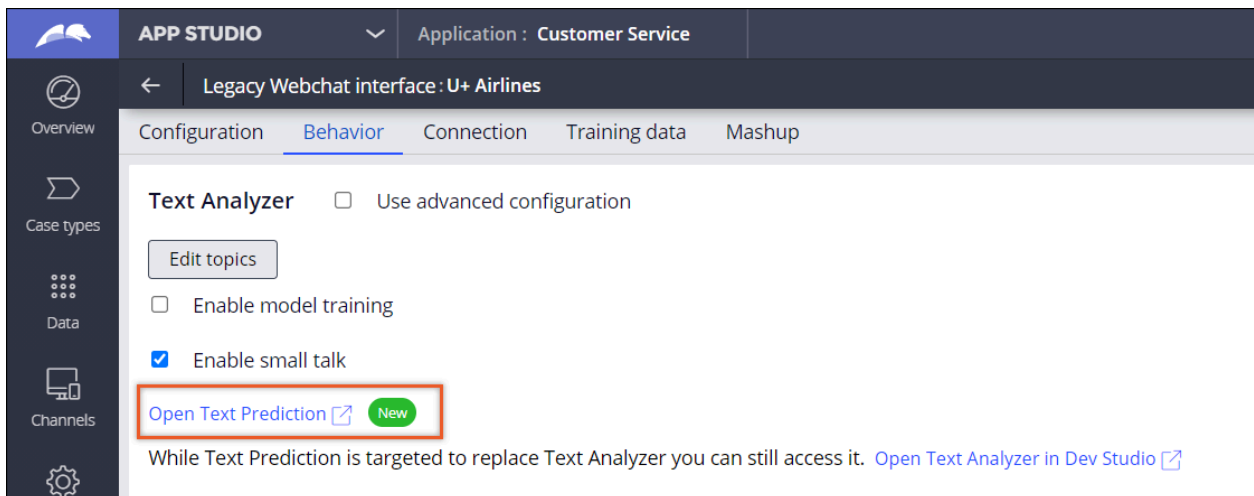
[Defining Email channel behavior \(on page 29\)](#)

Configuring a text prediction for an email bot

To improve the artificial intelligence of Pega Email Bot™ and its ability to detect the correct information from emails, you can configure a text prediction for an application. You can configure a text prediction to detect topics, entities, and sentiments in the email body, attachments, or both.

Tip: Text predictions are targeted to replace text analyzers. As a best practice, configure text analytics for your conversational channels by using text predictions instead of text analyzers. For more information, see [Text analysis concepts \(on page 21\)](#).

- In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
- In the navigation pane of App Studio, click **Channels**.
- In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
- In the Email channel, click the **Behavior** tab.
- In the **Text Analyzer** section, click **Open text prediction**.



Opening a text prediction from a channel

6. In Prediction Studio, configure the text prediction for the email bot.

For more information, see [Analyzing messages with text predictions \(on page 36\)](#).

Related information

[Adding a text analyzer for an email bot \(on page 36\)](#)

[Text analysis concepts \(on page 21\)](#)

Adding a text analyzer for an email bot

To improve the artificial intelligence of Pega Email Bot™ and its ability to detect the correct information from emails, you can configure one or more advanced text analyzers for an application. In this case, the text analyzers examine user input one by one, until the system finds a response.

For example, you can configure the first text analyzer to detect only entities or topics in the email body, another one to detect only entities or topics in the email attachments, and a third analyzer to detect entities or topics in both the email body and attachments.

Tip: Text predictions are targeted to replace text analyzers. As a best practice, configure text analytics for your conversational channels by using text predictions instead of text analyzers. For more information, see [Text analysis concepts \(on page 21\)](#).

Note: To learn more about troubleshooting natural language processing (NLP) issues in your email bots, see also [Troubleshooting NLP](#).

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. In the **Text Analyzer** section, select the **Use advanced configuration** checkbox.
In the advanced configuration mode, you configure the advanced text analyzers that use natural language processing (NLP) and adaptive and predictive analytics. If you do not use advanced configuration, you configure the default text analyzer that exactly matches user input to a response. For more information, see [Exploring text predictions and text analyzers \(on page 23\)](#).
6. In the **Text Analyzer** section, select a method for configuring a text analyzer:
 - To create a text analyzer, click **Add text analyzer**.
 - To edit an existing text analyzer, click the **Switch to edit mode** icon next to the text analyzer that you want to edit.
7. In the **Text analyzer type** list, select and configure a text analyzer:

Choices	Actions
Pega NLP	Configure an advanced text analyzer that uses the best approximate match by using advanced natural language processing (NLP) and artificial intelligence: a. Select whether to detect entities or topics in the context of the email body, attachments, or both. b. In the Text analyzer rule field, select or enter a text analyzer rule. c. To modify the text analyzer rule settings, click Open Text Prediction. For more information, see Building text analyzers (on page 23). d. To modify the text analyzer rule settings, click Open text analyzer in Dev Studio, make changes, and then in the lower-left of the page click Back to Pega App Studio.

Choices**Actions****iNLP**

Note: The iNLP text analyzer is associated with the default text prediction for the channel.

The option to modify the text analyzer rule is only available if you have access to Dev Studio.

For more information, see Building text analyzers (*on page*).

Configure an advanced intelligent NLP text analyzer / text prediction that uses adaptive analytics text analysis:

- a. Select whether to detect entities or topics in the context of the email body, attachments, or both.
- b. To modify the text prediction settings, click **Open Text Prediction**.
For more information, see Analyzing messages with text predictions (*on page*).
- c. To modify the text analyzer rule that underlies the text prediction, click **Open text analyzer in Dev Studio**, make changes, and then in the lower-left of the page click **Back to Pega App Studio**.

The option to modify the text analyzer rule is only available if you have access to Dev Studio.

For more information, see Building text analyzers (*on page*).

This type of analysis integrates text analytics with strategies, propositions, and interaction history to provide the context for making better next-best-action decisions. For more infor-

Choices	Actions
	mation, see Customizable Interaction API for text analytics (<i>on page</i>).

8. **Optional:** To enable subject analysis in email, select the **Enable subject analysis** checkbox.
For more information, see [Enabling subject analysis in email \(on page 41\)](#).
9. **Optional:** To enable email attachments analysis in email, in the **Analyze email attachments** list, specify how you want to analyze email attachments:
 - To always analyze email attachments, click **Always**.
 - To analyze email attachments based on a when rule, click **When**, and then in the adjacent field select or define a when rule.
 For more information, see [Enabling email attachments analysis during email triage \(on page 42\)](#).
10. Click **Submit**.
11. **Optional:** To add or configure more text analyzers for the email bot, repeat steps 6 (*on page 37*) through 10 (*on page 39*).
12. Click **Save**.

[Text analysis concepts \(on page 21\)](#)

[Defining Email channel behavior \(on page 29\)](#)

Defining topics for text analysis for an email bot

Define topics in Pega Email Bot™ so that the system determines the correct email subject by analyzing email content through natural language processing (NLP) and adaptive analytics text analysis models. Defined topics help the email bot detect and organize suggested responses and suggested cases in the system. For example, when users send a request for a bank loan to the email bot, the system correctly detects that the subject of the email relates to a bank loan. As a result, to help immediately address the issue during email triage, the email bot can automatically suggest a top-level bank loan case in your application to customer service representatives (CSRs).

You can configure multiple languages for the email bot by setting up text analysis for each language separately. This ensures that the system performs text analysis of email in the correct language that is detected. For more information, see [Creating an Email channel \(on page 26\)](#).

Note: All suggested cases that you define in the system are topics.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. **Optional:** To enable the advanced configuration mode, in the **Text Analyzer** section, select the **Use advanced configuration** checkbox.
In the advanced configuration mode, you configure the advanced text analyzers in the system that use NLP and adaptive analytics. If you do not use advanced configuration, you configure the default text analyzer that exactly matches user input to a response. For more information, see [Exploring text predictions and text analyzers \(on page 23\)](#).
6. In the **Text Analyzer** section, click **Edit topics** for the text analyzer that you want to edit.
7. In the **Edit topics** window, click **Add item**.
A topic is the general subject, the intent of an email that the system detects using text analysis.
8. If you configure multiple languages for the email bot, in the section in the **Edit topics** window, click a language.
The supported languages are displayed on a vertical tab. When you add a new topic or edit an existing topic, you can switch between the languages by clicking the name of a language.
9. In the table columns, specify a topic and matching criteria:
 - In the **Topic** field, enter the natural language processing topic for the email bot to use as the response. The NLP text analyzer determines the topic based on criteria that matches exactly, matches approximately, and or never matches the user input.
 - In the **Approximate match** field, to configure the email bot to look for approximate matches in the user input, enter words and phrases that you want to use, and then separate them with commas. For example, for an initial greeting responses, enter: hello, good morning.
 - In the **Must match** field, to configure the email bot to look for specific items in the email, enter words and phrases that you want to exactly match in the topic, and separate them with commas. For example, for initial greeting response, enter hi.
 - In the **Never match** field, enter comma-separated words or phrases to always ignore when matching for the topic. For example, for an initial greeting response to never match a goodbye greeting, enter bye.
10. **Optional:** To configure additional languages for the topic, repeat steps 8 [\(on page 40\)](#) through 9 [\(on page 40\)](#).
11. **Optional:** To add more topics for text analysis, repeat steps 7 [\(on page 40\)](#) through 10 [\(on page 40\)](#).
12. Click **Submit**.
13. Click **Save**.

The email bot uses text analysis to match content in the email body or attachments to a topic, a suggested case, or a suggested response.

[Text analysis concepts \(on page 21\)](#)

[Defining Email channel behavior \(on page 29\)](#)

Enabling subject analysis in email

To detect entities present in the subject of an email, enable subject analysis for Pega Email Bot™. With subject analysis, a text analyzer scans the subject field of an email for topics and entities. An email bot uses entities that are detected during text analysis in intelligent email routing, to determine the best action, for example, to open a case or send an automatic reply.

An entity is text that is detected by the system in an email and its attachments to help the email bot correctly respond to users. Entity text is comprised of proper nouns that fall into a commonly understood category, for example, a person, location, date, organization, or a ZIP code. With entity property mapping, the detected entities by the email bot are automatically assigned to case properties in a Pega Platform application.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. To enable advanced configuration mode, select the **Use advanced configuration** checkbox.
In the advanced configuration mode, you enable email subject analysis for the each advanced text analyzer separately.
6. If you use the advanced configuration mode for a text analyzer, in the **Text Analyzer** section, click the **Switch to edit mode** icon next to the text analyzer that you want to use for subject analysis.
7. Select the **Enable subject analysis** checkbox.
8. If you use the advanced configuration mode for a text analyzer, click **Submit**.
9. Click **Save**.

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

[Setting up entity property mapping \(on page 32\)](#)

[Enabling email attachments analysis during email triage \(on page 42\)](#)

[Defining Email channel behavior \(on page 29\)](#)



Enabling email attachments analysis during email triage

With Pega Email Bot™, you can automatically analyze the content of files that are attached to email to detect entities during email triage. Your email bot uses these entities and other information that is detected in text analysis in intelligent email routing, to determine the best action, for example, to open a case or send an automatic reply.

The system uses machine learning models that are trained to detect entities in email attachments. Pega Platform supports the following file types as attachments:

- Microsoft Word (**DOC/DOCX**)
- Open Document Format (**ODT**)
- Portable Document Format (**PDF**)
- Rich Text Format (**RTF**)
- Simple Text (**TXT**)

Note: You can also enable the analysis of content in email attachments based on a when rule.

- i** Only when the when rule holds true, the text analyzer scans the content of the attached files in an email.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. To enable advanced configuration mode, select the **Use advanced configuration** checkbox.
In the advanced configuration mode, you enable the analysis of email attachments for the advanced text analyzers.
6. If you use an advanced configuration mode for a text analyzer, in the **Text Analyzer** section, click the **Switch to edit mode** icon next to a text analyzer that you want to use for email attachments analysis.
7. In the **Text analyzer configuration** window, in the **Analyze email attachments** list, specify how you want to analyze email attachments:
 - To always analyze email attachments, click **Always**.
 - To analyze email attachments based on a when rule, click **When**, and then in the adjacent field select or define a when rule.
8. If you use an advanced configuration mode for a text analyzer, click **Submit**.
9. Click **Save**.



[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

[Enabling subject analysis in email \(on page 41\)](#)

[Defining Email channel behavior \(on page 29\)](#)

Enabling automatic language detection for text analysis

Configure the text analyzers used in Pega Email Bot™ to automatically detect the language of user emails. This feature ensures that the system supports multiple languages and improves the overall usability of the email bot.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon for your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. Select a method for opening a text analyzer rule:

Choices	Actions
For the default text analyzer	In the Text Analyzer section, click Open text analyzer rule. In the Text Analyzer section, click Open text analyzer in Dev Studio.
For an advanced text analyzer	a. In the Text Analyzer section, select the Use advanced configuration check box. b. Click the Switch to edit mode icon next to the text analyzer that you want to edit. c. In the Text Analyzer section, click Open text analyzer rule. In the Text Analyzer section, click Open text analyzer in Dev Studio.

6. In the text analyzer rule, click the **Advanced** tab.
7. In the **Language settings** section, click **Automatically detect language**.

8. **Optional:** To enable a fallback language if a language cannot be automatically detected by the system, select the **Enable fallback language if the language is undetected** checkbox, and then in the **Fallback language** list, select the fallback language.
9. Click **Save**.

[Text analysis concepts \(on page 21\)](#)

[Overriding the default language for an Email channel \(on page 93\)](#)

[Defining Email channel behavior \(on page 29\)](#)

Configuring intelligent email routing

To save time and ensure faster resolution of cases, configure your Pega Email Bot™ to react to user requests and automatically respond to emails. By creating conditions for intelligent email routing, the system can then automatically convert email content to a new interaction case for a user, add the email to a work queue, create a top-level case in an application, or send an automatic reply. For example, when you configure intelligent email routing to detect a credit card issue in the email subject and a customer email about a problem with their credit card transaction arrives, the email bot automatically routes the email to the correct operator and creates a top-level case to help track the issue in the system.

Use the customizable Interaction API for text analytics with your email bot to set up routing conditions based on a decisioning result, to determine the outcome of the interaction strategy. The *pyInteractionDF* data flow for the Interaction API performs natural language processing through the iNLP text analyzer for the email bot and feeds the outcome to the interaction strategy. The system captures the strategy outcome in the *pyDecisionResult* property for the *Data-pxStrategyResult* class. For more information, see Customizable Interaction API for text analytics (on page).

For relevant training materials and to test this feature on a virtual Pega instance, see the [Routing email based on content](#) module and the [Routing email based on content](#) challenge on Pega Academy.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. In the **Intelligent routing** section, in the **Process** list, select a processing mode:
 - To trigger only the first action in the list that matches the conditions, select **First matching action**.

All other actions in the list with matching conditions are not triggered.



- To trigger every *create case* and *automatic reply* action in the list that matches the conditions, and skip all *route to* actions, select **All matching actions**.

If no *create case* or *automatic reply* action in the list match the conditions, the system triggers only the first *route to* action that matches the conditions.

Selecting the routing processing mode enables the system to trigger only the first action or every *create case* and *automatic reply* action in the list with matching conditions.

6. In the **Intelligent routing** section, click **Add condition**.
7. In the **Action** list, select an action that you want to run automatically for email routing.
8. In the **Value** field, enter an email template, case name, operator, or work queue.
9. Define the validation conditions for a field on the form as the criteria for carrying out the routing action:

Choices	Actions
Detect email header content	- In the When list, select Header and the name of the email field that you want to validate, for example: Subject. - In the comparator list, select the test that you want to perform on the field, for example: Contains. - In the value field, enter or select a value to compare against the user input, for example: price.
Detect an entity - a word or phrase in an email	- In the When list, select NLP > Entity, and an entity name, for example: #Email. - In the comparator list, select the test that you want to perform on the field, for example: Is equal. - In the value field, enter a value to compare against the user input, for example: Bob@example.com.

Choices	Actions
Detect the topic - the subject matter of an email	a. In the When list, select NLP > Topic. b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value field, enter a value to compare against the user input, for example: Insurance Quote.
Detect a decisioning result - the outcome of the interaction strategy	a. In the When list, select NLP > Decisioning result. b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value field, enter a value to compare against the user input. The email bot can check whether the decisioning result is empty, is not empty, or contains a specified value as the outcome of the interaction strategy, for example, a postal code, or a company or product name. For more information, see ../dsd/da-portal/references/da-text-analytics-interaction-api-ref.dita.
Detect the positive, negative, or neutral sentiment of an email	a. In the When list, select NLP > Sentiment. b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value list, select what sentiment to compare against the user input, for example: Positive.

Choices	Actions
Detect the language of an email	a. In the When list, select NLP > Language. b. In the comparator list, select the test that you want to perform on the field, for example: Is equal. c. In the value list, select a language to compare against the user input, for example: French.
Use a when rule	a. In the When list, select Condition. b. In the field to the right of the When list, enter or select a when rule that you want to use as the condition for carrying out the action. For example, you can configure a routing condition to route the email to an operator <i>Andrew Sullivan</i> when the <code>pyIsInWorkList</code> rule is true.

Each routing condition specifies a single way that the email bot automatically responds to an email.

10. **Optional:** To add more automatic responses based on email header content, NLP text analysis of the email, or a when rule, repeat step 6 (on page 45) through 9 (on page 45) for other routing conditions.
 11. Specify how your email bot responds when you do not define any criteria or when the email content does not meet any of the criteria that you specified in step 9 (on page 45) by defining the default action:
 - a. In the **Otherwise** area, in the **Action** list, click an action to perform.
 - b. In the **Value** field, enter an email template, case name, operator name, or work queue.
 12. Click **Save**.
 13. Verify that the intelligent email routing works as expected.
- For more information, see [Testing email routing conditions \(on page 47\)](#).

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Testing email routing conditions

When you configure an Email channel, define routing conditions to improve the responses that Pega Email Bot™ provides to an email that you receive. Before training and moving the email bot natural language



processing (NLP) model to a production environment, you can use the test console to verify all the routing conditions for intelligent email routing.

You can test whether your email bot triggers the correct routing actions that you define for each routing condition case. In addition, you can verify sentiment detection.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. Click **Test**.
6. In the **Test routing conditions** window, verify a routing condition:
 - a. Challenge criteria for a routing condition by entering the parameters of a sample email.
 - b. Click **Test**.
 - c. In the **Outcomes** section, verify that the correct action occurs for the routing condition. You can follow several guidelines if the outcome of a routing condition is not correct. For more information, see [Troubleshooting email routing conditions \(on page 97\)](#).
 - d. Click **Close**.
7. **Optional:** To test all routing conditions, repeat step 6 ([on page 48](#)) for each routing condition.

[Defining Email channel behavior \(on page 29\)](#)

Training the model for the Email channel

To use Pega Email Bot™ in your application to seamlessly respond to user problems, train the system to recognize different user input in emails, such as help requests or other issues. When you train the text analytics model for the email bot, the system learns from training records, which improves the artificial intelligence algorithms and provides better responses to user input. For example, you train the model for the email bot and apply the changes to the model. Based on the detected topic, entities, sentiment, language in emails, and intelligent email routing, the email bot then runs automated tasks, such as starting a business case and sending an automatic reply to the user.

To ensure that the email bot learns from the training records and detects the correct information, such as topics and entities, apply the training changes to the text analytics model for the system. You apply changes to the model after you enable the training data recording, and then create and review the training records.



The system generates training data when a CSR:

- Spins off a business case by selecting a case from the suggested case or other cases list. For more information, see [Spinning off business cases for an email bot \(on page 66\)](#).
- Sends a reply by selecting a reply from the suggested reply or other replies list. For more information, see [Replying to customers by email for an email bot \(on page 59\)](#).

For relevant training materials, see the [Training the email bot](#) module on Pega Academy.

Note: You can also configure the system to train the text analytics model based on attachments in text and image files in addition to the email body. For more information, see [Enabling training the model based on email attachments \(on page 94\)](#).

1. Enable training of your text analytics model for the email bot.
For more information, see [Enabling the training data recording for an email bot \(on page 50\)](#).
2. Improve the email bot model by providing sample training records.
For more information, see [Creating training data manually for an email bot \(on page 50\)](#).
3. Update the training data by ensuring that the email bot detects the right information by using text analysis.
For more information, see [Correcting training data in an email bot \(on page 51\)](#).
4. **Optional:** To move the training data to another email bot, export and import the training records. This action makes the email bot easier to maintain and build in production, QA and development environments. For more information, see [Transferring training data to another email bot \(on page 53\)](#).
5. Make the email bot learn from the training data by building a text analytics model.
This action improves the artificial intelligence of the system. For more information, see [Applying changes to a text analytics model for an email bot \(on page 55\)](#).

Note: For more information about NLP and machine learning in email bots, see [Using email bots for natural language processing and machine learning](#).

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Enabling the training data recording for an email bot

To help Pega Email Bot™ make better email content predictions, enable the recording of training data in the system. You can then edit and classify the training data in the system by reviewing records and applying changes to the text analytics model. With these tasks, you ensure that the email bot detects the correct information for example, language, topics, and entities.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. On the Email channel configuration page, click the **Behavior** tab.
5. In the **Text Analyzer** section, activate the recording of training data:

Choices	Action
For the default text analyzer	a. Select the Record training data checkbox.
For an advanced configuration	a. In the text analyzer list, click the Switch to edit mode icon for a text analyzer. b. In the Text analyzer configuration window, select the Enable model training checkbox. c. Click Submit .

6. Click **Save**.

[Text analysis concepts \(on page 21\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Training the model for the Email channel \(on page 48\)](#)

Creating training data manually for an email bot

To ensure that Pega Email Bot™ interprets emails in the correct way, manually create training records in the system. Training records provide valuable information for the email bot that strengthens the artificial intelligence algorithms and improves the accuracy of your text analytics model. For example, you can add a training record that specifies a request for an insurance quote for a 1968 Ford Mustang GT. If you connect that record to a car insurance topic, the system learns to associate this information with a car insurance case.

You review the training data before applying the changes to the text analytics model.



The system generates training data when a CSR:

- Spins off a business case by selecting a case from the suggested case or other cases list. For more information, see [Spinning off business cases for an email bot \(on page 66\)](#).
- Sends a reply by selecting a reply from the suggested reply or other replies list. For more information, see [Replying to customers by email for an email bot \(on page 59\)](#).

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. On the Email channel configuration page, click the **Training data** tab.
5. In the toolbar, click **Add records**.
6. If you configure multiple languages for the email bot, in the **Creating new training record** window, in the **Language** list, select the language for the training record, for example, **English**.
7. In the **Topic** field, press the Down arrow key, and then select a name of a topic to which the training record relates, for example, **Car insurance**.
8. In the field area below the **Topic** field, enter the text for the training record.
The text must be a good representation of sample user input from which the email bot can learn to respond to emails.
9. Click **Create record**.

The system displays the training record in the list on the **Training data** tab. The **Review training data** section displays the training record contents with the detected entities highlighted. The **NLP analysis** section displays the detected language, topic, and entity types that are used in the training data.

10. **Optional:** To add more training records for the email bot, repeat steps 5 [\(on page 51\)](#) through 9 [\(on page 51\)](#).
11. Click **Save**.

[Training the model for the Email channel \(on page 48\)](#)

Correcting training data in an email bot

When you want to improve the ability of Pega Email Bot™ to detect topics, language, and entities, you can review and correct the training data in the system. By correcting the training data and rebuilding the text analytics model, you improve the artificial intelligence of the email bot and teach it to more accurately detect the desired information in emails. The system can then suggest the right business case or email response, based on the detected information. For example, you can correct training data related to a car



insurance request for an email bot, so that later the system correctly identifies a car insurance topic and entities related to a car make and model.

When you enable the recording of training data and the email bot receives an email, the system saves the email as a training record. (You can also manually create a training record that contains sample information for an email from a user). Once you correct the data detected in the training record, you mark the record as reviewed. The system then uses the reviewed training data to rebuild the improved text analytics model.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Training data** tab.
5. **Optional:** If you configure multiple languages for the email bot, to filter data records by a language, in the **Language** list, select a language.
6. In the list of training records, select a data record.

The **Review training data** section displays the detected entities, and the **NLP analysis** section displays the detected language, topic, and entity types for the training data record.

7. Correct the data for the training record:

Choices	Actions
Modify text	a. In the Review training data pane, click the More icon, and then click Edit. b. In the Update text window, edit the text for the training record, and then click OK.
Update the topic to be detected	a. In the Topic field in the NLP analysis section, press the Down arrow key. b. Select a more appropriate topic for the training record. For example, to correct a training record so that its intent relates to car insurance, select Car insurance.
Update the language to be detected	In the NLP analysis section, in the Language list, select a language.

Choices	Actions
Add new entities	a. In the Review training data section, in the data record content, highlight and right-click the text that you want to map to the new entity, and then click New entity. b. In the Create new entity window, in the Entity name field, enter a name for the entity, and then click Submit. For example, to create an entity for a car make, highlight the word Ford and enter Car Make as the new entity.
Update existing entities	In the Review training data section, in the data record content, highlight and right-click the text for an existing entity, and then click the name of another entity. For example, to make sure the car make in the text maps to the CarMake entity, highlight and right-click Ford, and then click #Car-Make.

8. **Optional:** To use this training record to improve the artificial intelligence of the email bot, in the **Review training data** section, click **Mark reviewed**.
9. **Optional:** To correct the data for additional training records for the email bot, repeat steps 6 (on page 52) through 8 (on page 53).
10. Click **Save**.

[Creating training data manually for an email bot \(on page 50\)](#)

[Training the model for the Email channel \(on page 48\)](#)

Transferring training data to another email bot

When you generate training data for Pega Email Bot™ that results in greater accuracy of topic and entity detection, you can reuse the information by transferring the training data to another email bot. Copying training data from one environment to another results in a quicker development of artificial intelligence of another email bot making the system easier to build since you do not have to create the text analytics model from scratch. For example, you can copy accumulated training data from a production environment



for an email bot to a development or a QA environment of another email bot. Before you can begin transferring data, you first start recording training data in the email bot.

Copying training data from one email bot environment to another involves first creating training data in a production environment, and then exporting the training data to a file. Next, you edit another email bot in a development or QA environment and import the saved training data from the file. You can export either only the reviewed training data for an email bot, or all of the training data. The training data that you export includes entities that the system added to the text analyzer but did not build in the model.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. Save training data from an email bot:
 - a. In the **Current channel interfaces** section, click the icon that represents your existing Email channel, for example, in a production environment.
 - b. In the Email channel, click the **Training data** tab.
 - c. Click the **More** icon above the list of training records, and then click **Export data**.

 **Note:** You can export data only if the email bot contains at least one training record.

- d. In the **Export data** window, select one of the following options:
 - To export only the reviewed data, click **Reviewed only**.
 - To export all data for the model, click **Reviewed and unreviewed**.
- e. Click **Export**.
- f. Click **Download** and save the training data as a **.zip** file.
4. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
5. In the navigation pane of App Studio, click **Channels**.
6. Reuse training data in another email bot:
 - a. In the **Current channel interfaces** section, click the icon that represents another existing Email channel, for example, in a development or QA environment.
 - b. In the Email channel, click the **Training data** tab.
 - c. Click the **More** icon above the list of training records, and then click **Import data**.
 - d. In the **Import data** window, click **Choose file** and select the **.zip** file that contains the training data that you exported in step 3.f (on page 54).

- e. Click **Import**.
- f. Update the text analytics model for the email bot by clicking the **More** icon next to the **Add records** button, and then clicking **Build model**.

[Training the model for the Email channel \(on page 48\)](#)

Applying changes to a text analytics model for an email bot

To add the reviewed training records to Pega Email Bot™ so that the system recognizes more information in emails, apply the modified records to the text analytics model. By applying these changes, you can build the model with additional classified training data, which improves how the email bot responds to emails by detecting the correct topic, entities, and language using text analysis.

The email bot records training data when a CSR creates a related business case manually for an email triage case or a CSR selects a reply from the suggested reply list.



Note: You can apply changes to the text analytics model only if training data exists for the Email channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. On the Email channel configuration page, click the **Training data** tab.
5. **Optional:** If you configure multiple languages for the email bot, to filter data records by a language, in the **Language** list, select a language.
6. In the list of training records, select the checkboxes for the records that you want to add to the model, and then click **Mark reviewed**.
7. Click the **More** icon next to the **Add records** button, and then click **Build model**.

[Training the model for the Email channel \(on page 48\)](#)

Triaging incoming emails

By manually triaging emails for the Pega Email Bot™, customer service representatives (CSRs) and other operators, such as managers, can work through multiple emails in the context of their application to address and resolve customer issues faster and more efficiently. Case workers triage emails in the Case Manager, Case Worker, or Email Manager portal. For example, if a customer emails a request for a car



insurance quote, a CSR can immediately spin off a business case related to the car insurance topic, forward the email to the company's financial adviser, and reply to the user by email, before resolving the triage case.

Email triage is the process of routing incoming emails to a user or to a work queue, directly creating a business case to handle the email, or taking actions in other ways to help address issues reported in the email. For more information, see [Email triage process \(on page 57\)](#).



Note: If you are experiencing issues with the Theme Cosmos design of your Email Manager portal, download the [Email Manager Cosmos component](#) from Pega Marketplace.

- Reply to the customer or other users for a triage case.
For more information, see [Replying to customers by email for an email bot \(on page 59\)](#).
- Forward emails to third-party stakeholders for a triage case to help address the requested issue more quickly.
For more information, see [Forwarding emails to other stakeholders for an email bot \(on page 61\)](#).
- Send a Pulse message to specific operators, or to everyone, to discuss the triage case.
For more information, see [Sending Pulse messages for an email bot \(on page 63\)](#).
- Assign a triage case to another user or a work queue.
For more information, see [Transferring triage cases \(on page 64\)](#).
- Create a business case that relates to a triage case.
For more information, see [Spinning off business cases for an email bot \(on page 66\)](#).
- Associate service requests with an existing triage case.
For more information, see [Associating service requests with a triage case \(on page 67\)](#).
- Detect duplicate or related business cases in the system.
For more information, see [Detecting duplicate or related cases for an email bot \(on page 68\)](#).
- Map a case to a duplicate or related case in the system.
For more information, see [Mapping to duplicate or related cases for an email bot \(on page 69\)](#).
- Link related spun-off or duplicate business cases to a triage case.
For more information, see [Linking related cases for an email bot \(on page 70\)](#).
- Mark a triage case as completed.
For more information, see [Resolving triage cases \(on page 71\)](#).

[Configuring intelligent email routing \(on page 44\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Email triage process

When Pega Email Bot™ receives an email, the email content is first examined by a text analyzer to detect topics, entities (such as people and places), sentiment, and language. Once the email bot completes the text analysis, the system applies automatic email routing based on the outcome of the text analysis and the routing rules that you define for the email bot.

The system creates a triage case as soon as the email arrives, even if none of the routing conditions defined for intelligent email routing hold true. Customer service representatives (CSRs) can then manually triage the email and choose the appropriate outcome. If a routing condition does hold true, then the system triggers the related action, for example, creating a business case in the system. You can view and process triage cases in the Case Manager or Email Manager portal just like regular cases, for example, by looking at the work queue or by clicking to get new work.

Note: By default, the email bot does not send automatic acknowledgment replies for incoming notification emails that the system created from service case requests. You can enable the sending of automatic acknowledgment replies for system notification emails by setting the *pyEnableSendAcknowledgment* when rule to true. This when rule has no effect on incoming emails from customers or other stakeholders for a triage case.

Email triage cases

An email triage case stores information from a received email. This information includes text analysis, attachments, and related users. Email triage cases help track information and resolve problems more quickly.

Unlike cases that are instances of a case type, email triage cases have the following characteristics:

- Email triage cases are intermediary objects that you manually triage.
- You do not control the definition of an email triage case as you would do for regular cases, because email triage cases are not instances of a case type.
- You do not create email triage cases.

You can view all triage cases generated by an email bot by using the Application Explorer. You can also perform manual triage of email triage cases generated by an email bot, from the Email Manager or Case Manager portals.

Email triage conversation

The email conversation details for an email bot session that the system saves in a triage case consist of emails from the customer, replies from the CSR, and replies from additional stakeholders to which a CSR forwarded the email, for example, vendors. The system displays the emails for a triage case in a list from the newest to the oldest.

When you configure an Email channel as an email bot, you can also automatically route the content in a received customer email message, or triage cases in the Case Manager or Email Manager that represent this email content. For example, you can transfer a triage case to another user or a work queue, or spin the triage case off as a specific business case.

CSRs can also save email drafts and simultaneously work on unrelated tasks for the email bot making the email triaging more efficient. To help CSRs quickly troubleshoot email issues, the email bot displays warning notifications during email triaging, if there are issues with the delivery of emails.

Note: For the system to display email delivery warning notifications for an existing Email channel that was created in previous versions of Pega Platform, save the existing channel in the system at least once.

To keep all related information about an issue in one place and to ensure quicker triage case resolution, CSRs working on an email triage case can also associate service requests with the triage case. Service requests are cases in customer service applications for a reported customer issue.

Email threads for a triage case

The system uses a secure threading mechanism to keep a conversation between a customer and a CSR in a separate email thread from the conversations with other stakeholders, for example, vendors. The system automatically embeds an encrypted security code in the body of each outbound email to help track any replies and forwarded emails from different stakeholders in the correct email thread for a triage case.

By default, the system hides the case identifiers in the email subject. However, if your organization requires that this information is shown, you can configure your Email channel to always display the case identifiers in the email subject. For more information, see [Creating an Email channel \(on page 26\)](#).

Note: If the encrypted security code is missing from a received email, the system automatically checks for case identifiers in the email subject.

[Configuring intelligent email routing \(on page 44\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

[Triaging incoming emails \(on page 55\)](#)

Replying to customers by email for an email bot

To ensure better responsiveness and quality of service in your organization, during the email triaging stage for Pega Email Bot™, a customer service representative (CSR) can use email to quickly respond to customers to help with requests or reported issues. For example, a CSR that receives an email asking about a car insurance quote can quickly reply to the customer, thanking them for their email and requesting any additional information necessary to generate their car insurance case.

The system displays the detected sentiment of each customer email in the thread (positive, negative, or neutral) to the right of the sender email address. As a result, the system shows the sentiment pattern for the entire customer email thread for a triage case. The sentiment shown helps CSRs to respond quicker and take immediate action for the triage case. For example, when CSRs notice that an email with a negative sentiment was detected, they can immediately send a reply to the user asking for more information, so that the CSR can process the request.

When you reply to a customer email, before sending your email, you can save a draft of the email and simultaneously perform other tasks, for example, work on other triage cases, forward the email to other stakeholders, or open an unrelated business case. The system displays information that you saved a draft of the email, including the total number of saved drafts.

Note:



- The system generates training data when a CSR selects a reply from the suggested reply or other replies list.
- For the system to display email delivery warning notifications for an existing Email channel that was created in previous Pega Platform versions, save the existing channel in the system at least once.

1. Open a portal by performing one the following actions:

- To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
- To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.



2. Open a portal by performing one the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view in the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. For an email to which you want to reply in an email thread, in the top right of the email, select how to reply for the triage case:
 - To reply only to the user who sent the email, click **Reply**.
 - To reply to all users for the email, click the **More** icon, and then click **Reply all**.

When the email bot receives a new email, the system displays a green indicator next to the email at the top of the thread.

The system displays emails in the thread in chronological order, from the most recent to the oldest.

5. **Optional:** To carbon copy or blind carbon copy more users in the reply, in the **Cc** or **Bcc** field, add more email addresses.
6. **Optional:** To include predefined text from a template in the reply, in the **Reply template** list, select a reply template.

The system displays the text for the selected reply template in the email text area.

7. In the text area, enter a reply to the user's email.
8. **Optional:** To attach files to the reply, click **Attach Files**, and then select a file.
9. **Optional:** To save a draft of your email, click **Save**.

You can save a draft of the email, perform miscellaneous tasks, and then proceed to step [10 \(on page 60\)](#) to send the email.

10. Click **Send**.

The system sends the email and displays the email reply at the top of the list in the current email thread.

11. If the system displays a warning notification after you sent the email, to learn more about the source of the error, expand the email thread with the notification, and then above the email subject line, click **recipient**.

Note: For the system to display email delivery warning notifications for an existing Email channel that was created in previous Pega Platform versions, save the existing channel in the system, at least once.

The system displays more details about the email delivery error.

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Forwarding emails to other stakeholders for an email bot

By forwarding emails, customer service representatives (CSRs) save time and make their job easier as they can resolve issues by exchanging information with third- stakeholders when needed. In addition, the system is made more secure by always keeping the forwarded email conversations in separate threads from the customer conversation, so that other stakeholders do not access classified information accidentally. For example, CSRs can address customer requests or reported issues in a triage case more efficiently by contacting additional stakeholders, such as third-party vendors, sellers, financial officers, or IT specialists. When a CSR receives a request for a car insurance quote, the CSR can forward the customer's email to the financial officer to immediately process the car insurance request.

The system displays the detected sentiment of each customer email in the thread (positive, negative, or neutral) to the right of the sender's name. As a result, the system shows the sentiment pattern for the entire customer email thread for a triage case. The sentiment shown helps CSRs to respond quicker and take immediate action for the triage case. For example, when CSRs notice that an email with a negative sentiment was detected, they can immediately escalate the reported issue in the email by forwarding the email to the correct stakeholders.

When you forward an email to other stakeholders, before sending your email, you can save a draft of the email and simultaneously perform other tasks, for example, work on other triage cases, forward the email to other stakeholders, or open an unrelated business case. The system displays information that you saved a draft of the email, including the total number of saved drafts.

Note:



- The email bot uses a secure threading mechanism to track emails in the system. As a result, when CSRs forward an email to stakeholders, the email subject will not include an identifier, for example, M-1234.
- For the system to display email delivery warning notifications for an existing Email channel that was created in previous Pega Platform versions, save the existing channel in the system at least once.

1. Open a portal by performing one the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.
2. Open a portal by performing one the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view in the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. Expand the email thread that lists an interaction between you and a customer or another stakeholder.

When the email bot receives a new email, the system displays a green indicator next to the email at the top of the thread.

The system displays emails in a thread in chronological order, from the most recent to the oldest.

5. For an email that you want to forward in an email thread, in the top-right corner of the email, click the **More** icon, and then click **Forward**.
6. In the **To** field, enter the user's email address to which you want to forward the current email.
7. **Optional:** To carbon copy or blind carbon copy additional users when you forward the email, in the **Cc** or **Bcc** field, enter more email addresses.
8. **Optional:** To include predefined text from a template in the forwarded email, in the **Reply template** list, select a reply template.

The system displays the text for the selected reply template in the email text area.

9. In the text area, add text for the email that you want to forward.

The system automatically appends the text from the initial email in the text area.

10. **Optional:** To attach files to the forwarded email, click **Attach Files**, and then select a file.

11. **Optional:** To save a draft of your email, click **Save**.

You can save a draft of the email, perform miscellaneous tasks, and then proceed to step [12 \(on page 63\)](#) to forward the email.

12. Click **Send**.

The system displays the **Forward** icon to the left of the email threads for forwarded content, to distinguish it from the main email thread with the customer.

The system sends the email and displays the forwarded email in a new email thread.

13. If the system displays a warning notification after you forwarded the email, to learn more about the source of the error, expand the email thread with the notification, and then above the email subject line, click **recipient**.

Note: For the system to display email delivery warning notifications for an existing Email channel that was created in previous Pega Platform versions, save the existing channel in the system, at least once.

The system displays more details about the email delivery error.

[Configuring the email bot to display a user name in emails \(on page 92\)](#)

[Configuring how the email bot handles a reply to forwarded emails \(on page 91\)](#)

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Sending Pulse messages for an email bot

Ensure that information reported in a triage case for an email bot is addressed in a timely manner by communicating with other application operators, by posting Pulse messages. A customer service representative (CSR) or other operator can send Pulse messages internally to specific users, or to everyone that has access to the email bot.

1. Open a portal by performing one of the following actions:

- To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
- To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.



2. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view in the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. In the right panel, click the **Pulse** tab.
5. In the **Pulse** section, select the recipients of the message:
 - To send a message to everyone, in the **Post** list, select **Post**.
 - To send a message to specific users only, in the **Post** list, select **Private post**, and then in the field below, press the Down arrow key and select user names.
6. In the **Start conversation** field below, enter the text for the Pulse message.
7. Click **Post**.

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Transferring triage cases

To ensure that a specific request in a user email is immediately forwarded to the correct person, a customer service representative (CSR) can transfer a triage case to another Pega Platform user, or a work queue. Transferring triage cases escalates the problem to the right person, streamlines the correct response, and efficiently processes user requests and reported issues. For example, a CSR can transfer an email request for a car insurance quote to an insurance manager, for immediate action.

1. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.
2. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view for the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.

4. In the toolbar above the text of the email message, click **Transfer**.
5. In the **Transfer** window, transfer the triage case to one of the following users:

Choices	Action
Operator	a. In the Transfer to list, select Operator. b. To select an operator from your own work group, click Operator from the work group 'My team' To select an operator from outside of your work group, click Operator from the system. c. In the Operator field, press the Down arrow key, and then select an operator. d. Optional: To add a note to the transfer, in the Audit note field, enter a message about the transfer action for the next operator. e. Click Transfer.
Reporting manager	a. In the Transfer to list, select Reporting manager. b. Optional: To modify the default transfer note, in the Audit note field, edit the text describing the transfer action for the reporting manager. c. Click Transfer.
Work queue	a. In the Transfer to list, select Work queue. b. In the Work queue field, press the Down arrow key, and then select a work queue. c. Optional: To add a note to the transfer, in the Audit note field, enter a message about the transfer action for the work queue. d. Click Transfer.

Choices	Action
Work group	- In the Transfer to list, select Work group. - In the Work group field, press the Down arrow key, and then select a work group. Optional: To add a note to the transfer, in the Audit note field, enter a message about the transfer action for the work group. - Click Transfer.
Work group manager	- In the Transfer to list, select Work group manager. Optional: To modify the default transfer note, in the Audit note field, edit the text describing the transfer action for the work group manager. - Click Transfer.

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Spinning off business cases for an email bot

To immediately address a user request sent by email, you can spin off a related business case for triage. By performing this action, you ensure that your application automatically saves the information described in a business case, and marks it for immediate action. As a result, the system encapsulates user requests inside a business case life cycle, for quicker response and resolution of the reported issue. For example, a customer service representative (CSR) can immediately save a user request for a car insurance quote as an Insurance Quote business case.



Note: The system generates training data when a CSR spins off a business case by selecting a case from the suggested case or other cases list.

1. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.
2. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view in the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. In the right panel, click the **Details** tab.
The **Create case** section displays a list of suggested business cases for the triage case.
5. In the **Create case** section, click the name of a business case that you want to spin off for triage.
6. In the **Create case name** section of the **Description** field, enter a short description of the triage case, to save in the business case.
7. In the **Label** field, enter the name of the operator that is triaging the case.
8. Click **Create case**.

A message displays, stating that a new business case has been created, for example: New "Car Insurance" C-2723 was created.

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Associating service requests with a triage case

Ensure that customer service representatives (CSRs) have access to all of the relevant information in one triage case for customers that request help or provide additional details about the same issue multiple times. The system creates service requests from incoming customer emails. By associating service requests with the same triage case, CSRs have more control over the details in the reported information, making their job easier and helping to resolve the triage case quicker. For example, a customer moves to another city and sends an email with their updated mailing address. The customer then realizes that they forgot to include their new postal code. As a result, the customer sends a follow-up email in which they specify their new postal code. The system creates two separate service requests about the address change from the incoming emails. To keep all relevant information in one place in the system, a CSR can then associate both related service requests with a single triage case that describes the request for the customer.



1. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.
2. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view for the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. In the toolbar above the text of the email message, click **Associate**, and then configure associations between cases:
 - To associate a service request with the triage case, select the check box next to the service request name and identifier.
 - To disassociate a service request from the triage case, clear the check box next to the service request name and identifier.

The system displays a window with a list of service requests that are related to the triage case.

5. Click **Save**.

The system displays associated service requests in the panel on the right, in the **Related cases** section.

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Detecting duplicate or related cases for an email bot

Customer service representatives (CSRs) can quickly detect duplicate or related business cases during the triage process, to improve the general flow of information reported by a user in an email. As a result, all information reported about an issue is available from one place in the system. A CSR can map or link duplicate or related business cases in the Email Manager or Case Manager portal.

1. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.
2. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view in the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. In the right panel, click the **Details** tab.

The **Create case** section displays a list of suggested business cases for the triage case.

5. In the **Create case** section, click the name of the new business case you want to spin off for the triage case.

If the system finds duplicate or related business cases, the **Potential duplicate cases have been found** window displays a list of the duplicate or related cases.

6. **Optional:** To skip the comparison of the duplicate case and just create a new business case, click **Skip and create new case**.

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Mapping to duplicate or related cases for an email bot

Customer service representatives (CSRs) can overwrite information in case fields in a duplicate or related business case, with information from the new case. By mapping information to cases, you ensure that the same relevant information is available in all cases describing the same issue or business request. For example, CSRs can overwrite the account identifier and credit card number properties in a duplicate business case, so that the same information is available in the new business case that they spin off.

1. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.



2. Open a portal by performing one of the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view in the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. In the right panel, click the **Details** tab.

The **Create case** section displays a list of suggested business cases for the triage case.

5. In the **Create case** section, click the name of the new business case that you want to spin off for the triage case.

If the system finds duplicate or related cases, the **Potential duplicate cases have been found** window displays a list of duplicate or related cases.

6. In the **Potential duplicate cases have been found** window, in the row for a duplicate or related case, click **Compare**.
7. In the compare with existing case window, select the checkboxes next to the case properties for the new business case whose values you want to overwrite in an existing duplicate or related case.
8. Click **Mark as duplicate**.
9. **Optional:** To map properties to another duplicate case, repeat steps [6 \(on page 70\)](#) through [8 \(on page 70\)](#).

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Linking related cases for an email bot

By linking related business cases in the Email Manager or Case Manager portal to a triage case, customer service representatives (CSRs) ensure that all of the relevant information from separate cases is easily accessible from one place. As a result, CSRs have access to all required information and can respond to user issues or requests more efficiently. A related business case is automatically linked in the system to the triage case when you spin off this business case or mark it as a duplicate.

- Spin off a business case from an email triage case.
For more information, see [Spinning off business cases for an email bot \(on page 66\)](#).
- Mark as duplicate related business cases for a new business case.
For more information, see [Mapping to duplicate or related cases for an email bot \(on page 69\)](#).



[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Resolving triage cases

Once a customer service representative (CSR) carries out all of the necessary actions during the email triage process, they can mark the triage case as completed by resolving it. By properly resolving triage cases, the CSR ensures that Pega Email Bot™ manages user emails more efficiently, and that only the currently unresolved requests from users are tracked. For example, a CSR addresses a request for a car insurance quote by sending an email reply to the user, spinning off a related business case, and then resolving the triage case in the system.

1. Open a portal by performing one the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Email Manager portal, in Dev Studio, click **Launch web interface**, and then click **Email Manager**.
2. Open a portal by performing one the following actions:
 - To open the Case Manager portal, in App Studio, click **Case Manager**.
 - To open the Email Manager portal, in App Studio, click **Case Manager**, and then click **Email Manager**.
3. Open a triage case by performing one of the following actions:
 - To edit a triage case in the Case Manager portal, in the dashboard view in the list of cases, click a case name.
 - To edit a triage case in the Email Manager portal, in the list of emails, click an email.
4. In the toolbar above the text of the email message, click **Resolve**.

The status of the triage case changes to resolved.

[Email triage process \(on page 57\)](#)

[Triage incoming emails \(on page 55\)](#)

Setting up work queues for email processing

You can improve how customer service representatives (CSRs) triage emails in the Email Manager portal, by optionally configuring work queues in Email Manager. By doing this, you ensure that CSRs and other operators in different roles all have easy access to the same triage cases for user issues and requests from one place. For example, to respond to user inquiries, CSRs can use a work queue listing user requests related to car insurance, to quickly assign triage cases to themselves or to other operators.



1. Create work queues in the Pega Platform application.
For more information, see [Creating a work queue](#).
2. Assign CSRs or other operators to the work queues.

CSRs or other operators can select triage cases from the work queues in the Email Manager portal.

[Email triage process \(on page 57\)](#)

Viewing reports for the Email channel

Display information about the number of received emails, triaged cases, and created business cases in Pega Email Bot™. The data shown in the built-in email channel reports helps you to later adjust machine learning models to attain greater automation of the system. For example, if the statistics in a report show that more emails were manually triaged by CSRs than automatically triaged by the system, you can take steps to improve the artificial intelligence and intelligent email routing, so that automatic triaging occurs more frequently.

By default, customer service representatives (CSRs) and other users can use the Email Manager portal to view reports about the email bot. You can also customize the system to display the reports as a plug-in dashboard in other portals, for example, the Case Manager portal. For more information, see [Adding the email bot reports dashboard to other portals \(on page 90\)](#).



Note: The built-in reports dashboard is only displayed in the portal if you define at least one Email channel in the system.

1. In Dev Studio, click **Launch web interface**, and then click **Email Manager**.
2. In App Studio, click **Case Manager**, and then click **Email Manager**.
3. In the Email Manager portal, in the upper-right of the page, click **Analytics**.
4. In the **Email reports** section, in the **Channel** list, select the Email channel for which you want to view reports.
5. In the **View for** list, select one of the following options:
 - **Today** - display data in the reports only for today.
 - **Yesterday** - display data in the reports only for yesterday.
 - **Last 7 days** - display data in the reports from the last 7 days.
 - **Custom** - display data in the reports for a selected date range.
6. If you select the **Custom** view in the previous step, in the **From** and **To** fields, select the start date and the end date for the period.
7. In the **Email account** list, select the email account for which you want to view reports.

When you create an Email channel, you can define multiple email accounts for the email bot. For more information, see [Creating an Email channel \(on page 26\)](#).

8. **Optional:** To display data about automatically created business cases and manually created business cases only for one case type, in the **Case overview for** report, in the list, select a case type name.

By default, the **Case overview for** report displays data for all the case types.

[Enabling reports for the email bot \(on page 89\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Built-in reports for the email bot

In the Email Manager portal, you can view built-in reports for Pega Email Bot™ as line or pie charts. The three charts in each report display valuable information in real time about received emails, actions for triage cases, and created business cases. By visualizing how your email bot is performing in the system, you can improve how the email bot responds to users. For example, you can adjust the machine learning models to achieve greater automation in the system.

An email bot includes three built-in reports by default. You can select the email channel for which you want to display the reports, select the email account for the channel for which you want to view data, and select to display information for a specific time span: current day, last day, last 7 days, or a specific date range.

You can also customize the system to display the reports as a dashboard in other portals, for example, the Case Manager portal, and create custom reports with additional information that is not part of the built-in reports.

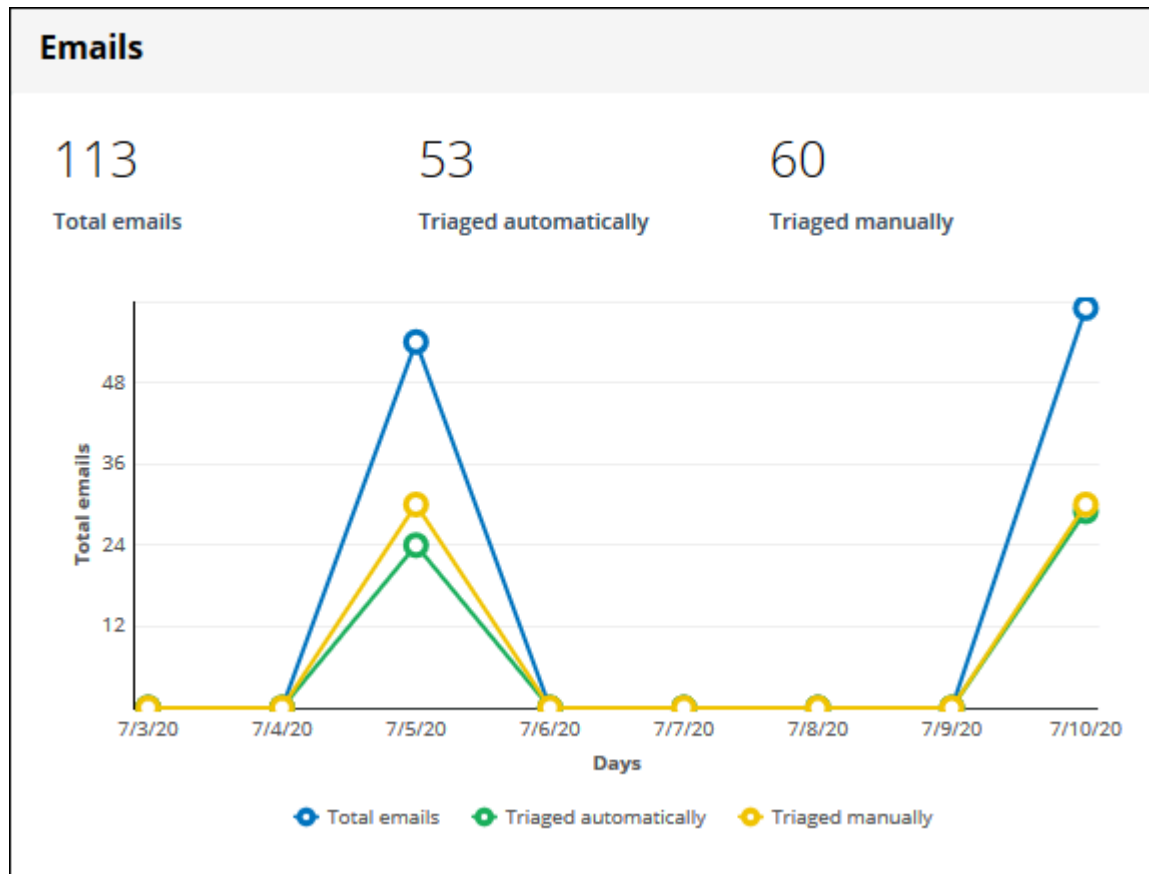
Email routing actions chart

This line chart describes information about the total number of emails that were triaged in the system, the number of emails that were only automatically triaged by using intelligent routing, and the number of emails that were manually triaged by customer service representatives (CSRs). The chart helps you visualize the automation level in your email bot by comparing the number of emails that were automatically triaged, and the number of emails manually triaged by CSRs. For example, if the statistics in the chart show that more emails were manually triaged by CSRs than automatically triaged by the system, you can take steps to improve the artificial intelligence and intelligent email routing, so that automatic triaging occurs more frequently.



Note: You can hover over any point in the chart to display details about the number of emails that were triaged for that day.

The following figure shows the email routing actions chart with sample data:



An email routing actions chart displaying data in the Email Manager portal for 384 total emails.

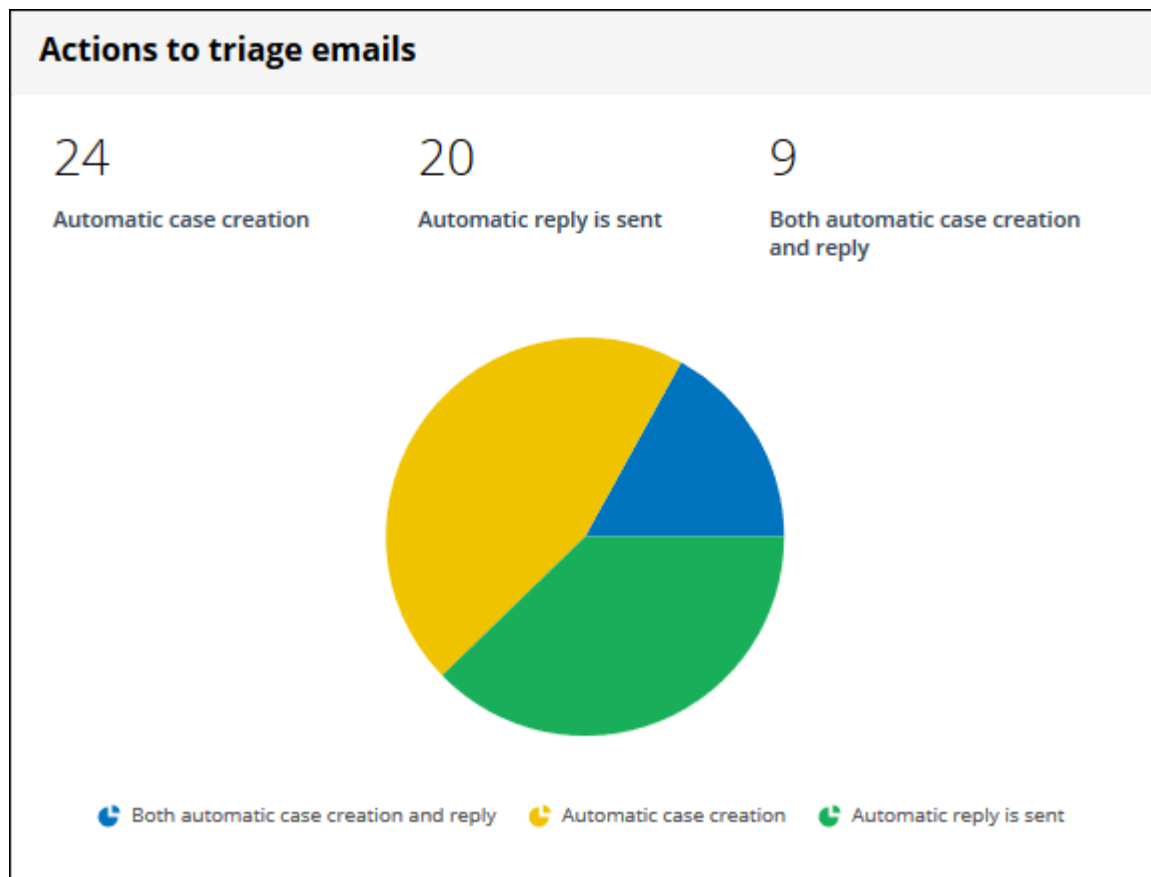
Email triage actions chart

This pie chart describes information about actions that the system performs when triaging emails. The chart displays the number of triage cases for which the email bot automatically created a business case, automatically sent an email reply, or both automatically created a business case and sent an email reply. The chart helps you to visualize and compare the frequency of response actions performed by the system. For example, if the chart statistics show that the email bot automatically creates business cases more often than sending automatic replies, it means that the system correctly processes email requests and assigns this information to new business cases.



Note: You can hover over each section in the pie chart to display information about the percentage share of a triage action out of the three types.

The following figure shows the email triage actions pie chart with sample data:



An email triage actions pie chart displaying data in the Email Manager portal for 73 triaged actions.

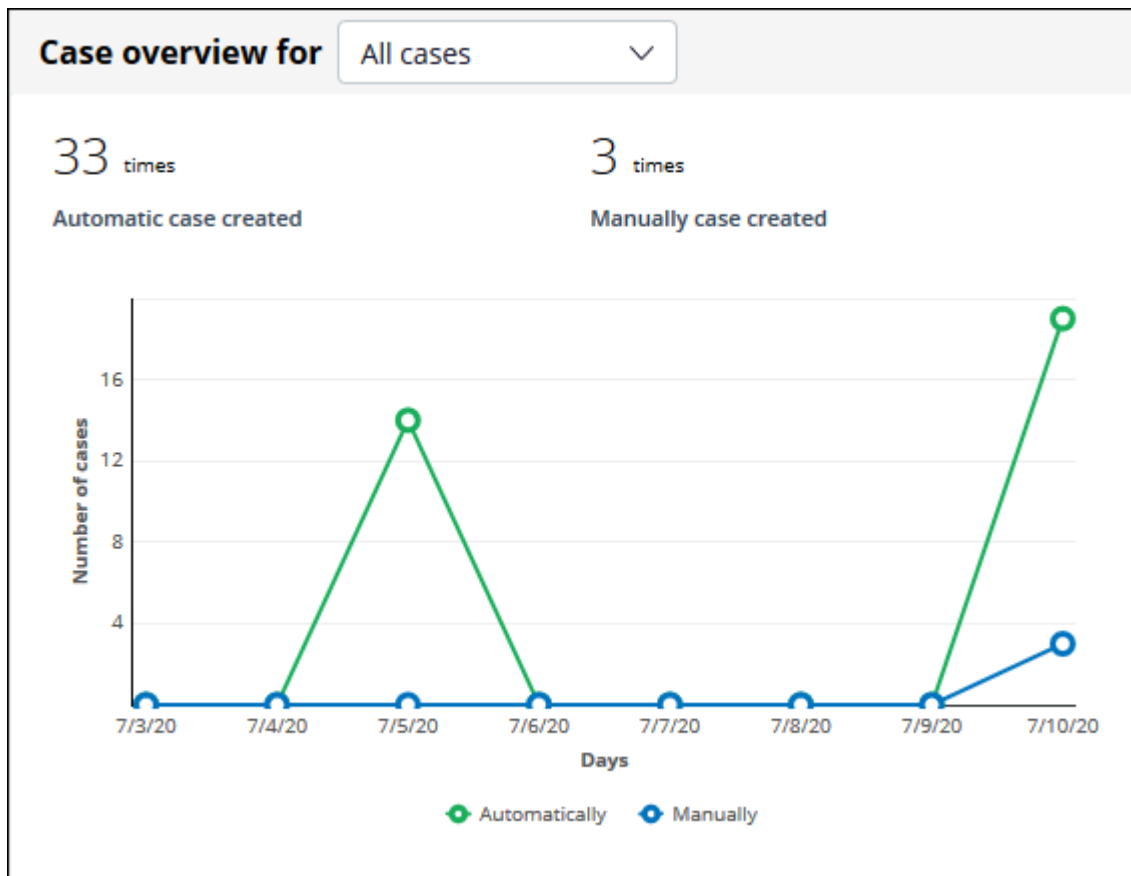
Email case overview chart

This chart describes case information about the number of times that a business case was created automatically by the email bot, or manually by CSRs. By using a drop-down list in this chart, you display data for a single case type or for all the case types for your email channel. The chart helps you to visualize whether the system more often automatically created business cases or the CSRs manually created business cases, more often. For example, if the statistics on the chart show that the CSRs created cases manually more frequently than when they were created by the email bot, you can take steps to improve the artificial intelligence and automation to help the system create business cases more often without any CSR interaction.



Note: You can hover over any point in the chart to display details about the number of business cases created automatically and manually for that day.

The following figure shows the email case overview chart with sample data:



An email case overview chart displaying data in the Email Manager portal for 51 business cases created automatically or manually.

[Viewing reports for the Email channel \(on page 72\)](#)

[Enabling reports for the email bot \(on page 89\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Customizing the Email channel

Carry out additional configuration of Pega Email Bot™ to help fine-tune its behavior and ensure that the system works as expected. As a result, you improve the overall experience of how users interact with your application. For example, you can customize the look and feel of the emails sent by the system or customer service representatives (CSRs) to conform to your organization's unique branding and requirements.

- Configure replies and forwarded emails by performing the following tasks:
 - Ensure that the system displays images inline and in a rich text format in triage cases.

For more information, see [Enabling inline images in email triage cases \(on page 77\)](#).

- Customize additional processing for an email reply to forwarded emails.

For more information, see [Configuring how the email bot handles a reply to forwarded emails \(on page 91\)](#).

- Customize the system to display user names instead of email addresses in emails.

For more information, see [Configuring the email bot to display a user name in emails \(on page 92\)](#).

- Create additional outbound email templates that you can select during the Email channel configuration so that emails sent by customer service representatives (CSRs) have a consistent look-and-feel.

For more information, see [Creating outbound email templates \(on page 85\)](#).

- Customize email bot reports by performing the following tasks:
 - Enable the reports dashboard and the gathering of data for the built-in and custom email bot reports.

For more information, see [Enabling reports for the email bot \(on page 89\)](#).

- Insert the reports dashboard in the Case Manager or Case Worker portals.

For more information, see [Adding the email bot reports dashboard to other portals \(on page 90\)](#).

- If you plan to support multiple languages in your application or one language other than English, you can override the default language of a new Email channel.

For more information, see [Overriding the default language for an Email channel \(on page 93\)](#).

[Pega Email Bot overview \(on page 5\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Conversational channels \(on page 4\)](#)

Enabling inline images in email triage cases

Update the configuration for your Pega Email Bot™ so that the system can display images inline in the rich text format for triage cases. By showing images in rich text format, you reap the following benefits: make the information more presentable, improve the overall user experience, and increase the performance of the system for users and customer service representatives (CSRs).

When the email bot generates triage cases from user emails, the system might not display images in the email body inline in the rich text format. To fix this issue, configure the auto-generated service email rule for your email bot to store the email body as an HTML attachment. Also, configure the email listener rule for your Email channel so that it does not embed inline images.

When you perform the following configuration steps, you ensure that the system saves the email body and its inline images as attachments. At run time, to increase performance, the system then changes these attachments to the triage case to display as inline images in rich text format.



Note: For security reasons, by default, in triage cases, the system does not display images that have an external URL. For example, images that have an external URL in emails might be a target for SQL injection attacks. To display such images in triage cases, define the external domains where the images are stored in the *pyAllowedDomainsForInlinelImages* data transform rule.

1. Switch to Dev Studio.
2. Configure the email listener rule not to embed inline images into the HTML:
 - a. In the navigation pane of Dev Studio, click **Records**.
 - b. Expand the **Integration-Resources** category, and then click **Email Listener**.
 - c. In the list of email listener rule instances, open the rule used by your email bot by clicking the rule's name.
To find the email listener that your email bot uses, edit the email listener rule. When you edit the rule, on the **Properties** tab, in the **Email account** section, ensure that the email account in the field is the same account that you defined for your Email channel on the **Configuration** tab, in the **Email handling** section, in the **Email accounts** list. For more information, see *Creating an email listener (on page)* and *Exploring email configurations (on page)*.
 - d. In the email listener rule, click the **Process** tab.
 - e. In the **Data options** section, clear the **Embed data for inline images into HTML** check box.
 - f. Click **Save**.
3. Configure the service email rule to store the email body as an HTML attachment:
 - a. In the navigation pane of Dev Studio, click **Records**.
 - b. Expand the **Integration-Services** category, and then click **Service Email**.
 - c. In the list of service email rule instances, to open the rule for your email bot, click the rule name.
To find the service mail rule that your email bot uses, edit email listener rule for your email bot. When you edit the rule, on the **Properties** tab, in the **Listener properties** section, check the name of the service mail rule in the **Service package** field. For more information, see

Creating a Service Email rule (on page) and Exploring email configurations (on page).

- d. In the service email rule, click the **Request** tab.
- e. In the **Message data** section, in the **Handle HTML content** list, select **As HTML attachment**.
- f. Click **Save**.

4. **Optional:** To hide inline images from also displaying as attachments in triage cases:

- a. In the navigation pane of Dev Studio, click **App**, and then search for the *Data-WorkAttach-File* class.
- b. Expand the **Decision > When** section for the *Data-WorkAttach-File* class, and then click **pyDisplayEmbeddedFileAsAttachment**.
- c. In the *pyDisplayEmbeddedFileAsAttachment* when rule, in the **Set conditions** section, in the first list in the row select **Always**, and then in the second list, select **is false**.
- d. Save your changes by performing one of the following actions:
 - To save the rule that belongs to a locked ruleset to an isolated sandbox so that you can test your changes, click **Private edit**.

For more information, see Performing a private edit (on page).

 - To save the rule to a ruleset, click **Save as**.

By default, the system displays inline images in triage cases also as attachments.

5. **Optional:** To display externally referenced images in triage cases, perform the following steps:

- a. Search for the *pyAllowedDomainsForInlineImages* data transform rule, and open the rule.
- b. On the **Definition** tab, click the **Add a row** icon.
- c. In the **Action** column for the row, select **Set**.
- d. In the **Target** column for the row, enter: `pxResults(<APPEND>).pyNote`
- e. In the **Source** column for the row, enter the domain where you store an external image in double quotation marks.
- f. **Optional:** To define additional domains where you store your external images, repeat steps 5.b (on page 79) through 5.e (on page 79).
- g. Click **Save**.

6. **Optional:** To encrypt sensitive information embedded in inline image HTML tags for a triage case, add URL mappings in the system.

For more information, see [Adding URL mappings to secure inline images for triage cases \(on page 80\)](#).

7. Switch back to App Studio.

[Creating an Email channel \(on page 26\)](#)

[Troubleshooting the email bot \(on page 95\)](#)

[Customizing the Email channel \(on page 76\)](#)

Adding URL mappings to secure inline images for triage cases

Ensure that inline images displayed in email triage cases are more secure for Pega Email Bot™. By adding URL mappings to your application, sensitive data embedded in inline image attributes for a triage case is encrypted and displayed by the system as hash code. The URL Mappings rule defines simple URL patterns that you can use in your application's work items, for example, email triage cases and its inline images. For example, after you add URL mappings and you then enable web developer tools for your browser, the attribute information for a triage case inline image HTML tags is encrypted and displayed by the system as hash code.



Note: Add URL mappings to the top-most *pyDefault* rule in your application, no matter if an administrator enabled or disabled URL obfuscation dynamic system settings in your system. URL obfuscation is an encoding technique that scrambles the characters in a URL or query string as a security measure, to make the URL contents less easily interpreted by people or HTTP sniffers. For more information, see [Understanding dynamic system settings \(on page 80\)](#) and [Creating a dynamic system setting \(on page 81\)](#).

1. Switch to Dev Studio.
2. In the navigation pane of Dev Studio, click **Records**.
3. Expand the **Technical** category, and then click **URL Mappings**.
4. In the list of URL mappings rule instances, open the top-most rule for your application: *pyDefault*.
5. Define the *getEmailImgByHash* URL alias:
 - a. In the **Cases** section, click **Add URL alias**.
 - b. In the **Define URL Mapping** window, in the **Identifier** field, enter: `GetEmailImgByHash`
 - c. Clear the **Map path elements for URL generation** checkbox.



- d. In the list in the **Path element type** column, select **Constant**, and then in the field in the **Value** column, enter: `getEmailImgByHash`
- e. Click **Add path element**, and then in list in the second row in the **Path element type** column, select **Parameter**.

The field in the **Value** column in the second row displays `{param1}`.

- f. Click **Add path element**, and then in list the third row in the **Path element type** column, select **Parameter**.

The field in the **Value** column in the third row displays `{param2}`.

- g. Click **Next**.

The following figure shows the URL mapping settings for the `getEmailImgByHash` URL alias:

Define URL mapping

Identifier *

GetEmailImgByHash

Build resource path ?

getEmailImgByHash/{param1}/{param2}

☐ Map path elements for URL generation

Path element type	Value	
Constant	getEmailImgByHash	
Parameter	{param1}	
Parameter	{param2}	

[+ Add path element](#)

Next >>

The settings for the new `getEmailImgByHash` URL mapping

- h. In the **Define processing activity** window, in the **Class** field, enter: `Work-Channel-Triage`
- i. In the **Activity** field, enter: `pyGetImageForDisplay`
- j. Click **Add parameter**, and then in the **Parameter** field, enter: `pyId`

- k. In the **Value** field, enter: {param2}
- l. Click **Add parameter**, and then in the **Parameter** field, enter: hash
- m. In the **Value** field, enter: {param1}
- n. Click **Add parameter**, and then in the **Parameter** field, enter: cid
Leave the **Value** field for the *cid* parameter empty.

The following figure shows the activity mapping for the *getEmailImgByHash* URL alias:

Edit processing activity [Close]

Activity mapping [Help]

Resource path
getEmailImgByHash/{param1}/{param2}

Class★
Work-Channel-Triage

Activity★
pyGetImageForDisplay [Target]

Parameter	Value	
pyId	{param2}	[Delete]
hash	{param1}	[Delete]
cid	New value	[Delete]

[+ Add parameter](#)

<< Back Finish

The settings for the *getEmailImgByHash* activity mapping

- o. Click **Finish**.

The list in the **Cases** section displays the *getEmailImgByHash* URL mapping.

6. Define the *getEmailMagByCID* URL alias:

- a. In the **Cases** section, click **Add URL alias**.
- b. In the **Define URL Mapping** window, in the **Identifier** field, enter: GetEmailImgByCID
- c. Clear the **Map path elements for URL generation** checkbox.
- d. In the list in the **Path element type** column, select **Constant**, and then in the field in the **Value** column, enter: getEmailImgByCID

- e. Click **Add path element**, and then in list the second row in the **Path element type** column, select **Parameter**.

The field in the **Value** column in the second row displays {param1}.

- f. Click **Add path element**, and then in list the third row in the **Path element type** column, select **Parameter**.

The field in the **Value** column in the third row displays {param2}.

- g. Click **Next**.

The following figure shows the URL mapping settings for the *getEmailImgByCID* URL alias:

Define URL mapping

Identifier ★

getEmailImgByCID

Build resource path ?

getEmailImgByCID/{param1}/{param2}

☐ Map path elements for URL generation

Path element type	Value
Constant	getEmailImgByCID
Parameter	{param1}
Parameter	{param2}

[+ Add path element](#)

Next >>

The settings for the new *getEmailImgByCID* URL mapping

- h. In the **Define processing activity** window, in the **Class** field, enter: Work-Channel-Triage

- i. In the **Activity** field, enter: pyGetImageForDisplay

The system displays three rows in the **Parameter** column for the *pyID*, *hash*, and *cid* parameters.

- j. In the row for the *pyID* parameter, in the **Value** field, enter: {param2}
- Leave the **Value** field for the *hash* parameter empty.

- k. In the row for the *cid* parameter, in the **Value** field, enter: {param1}

The following figure shows the activity mapping for the *getEmailImgByCID* URL alias:

Define processing activity

Activity mapping ?

Resource path
getEmailImgByCID/{param1}/{param2}

Class★
Work-Channel-Triage

Activity★
pyGetImageForDisplay

Parameter	Value
pyId	{param2}
hash	New value
cid	{param1}

+ Add parameter

<< Back Finish

The settings for the *getEmailImgByCID* activity mapping

- l. Click **Finish**.

The list in the **Cases** section displays the *getEmailImgByCID* URL mapping.

7. Save your changes by performing one of the following actions:

- To save the rule that belongs to a locked ruleset to an isolated sandbox in order to test your changes, click **Private edit**.

For more information, see *Performing a private edit (on page 77)*.

- To save the rule to a ruleset, click **Save as**.

8. Switch back to App Studio.

[Enabling inline images in email triage cases \(on page 77\)](#)

[Creating an Email channel \(on page 26\)](#)

[Customizing the Email channel \(on page 76\)](#)

Creating outbound email templates

To ensure better and customized branding in Pega Email Bot™ when interacting with customers, you can specify how sent emails look by creating new templates to fit the needs and requirements of your organization. As a result, automatic acknowledgment emails from the email bot or the emails that customer service representatives (CSRs) send always have a consistent look and feel that corresponds to your company branding.

To define a new outbound email template with a `<template_name>` name, you can take two different approaches. You can create a copy of one of the built-in templates that is closest to the desired end result by first finding the four rules for the outbound email templates by searching for `_Classic`, `_Cobalt`, or `_Clear` in the system. You then copy the four rules keeping them in the same class, removing the `py` prefix and appending `<template_name>`. Pega recommends this approach of copying the four rules and updating them to fit the requirements of your organization. Alternatively, you can create all four rules from scratch. The four rules for the new outbound email template you create must include the `<template_name>` suffix so that later you can select the new template during Email channel configuration.

If you create all four rules from scratch, to use the secure built-in threading mechanism for tracking email replies in the correct email thread for a triage case, add the following tag in the HTML source for your new `EmailFormat_<template_name>` rule:

```
<pega:include name="pyEmailThreadIdentification" type="Rule-HTML-Paragraph"></pega:include>
```

Because you define this HTML tag in the rule, the system embeds an encrypted security code in the email body to help track email replies in the correct email thread for a triage case. The built-in templates already include this HTML tag in the rule. For more information, see [Email triage process \(on page 57\)](#).

Note: If you update from Pega Platform version 8.5 or earlier, in which you configured email channel, for the initial acknowledgment email, update the service method for your email listener rule. On the **Response** tab for the service email rule you specify the rule that defines the structure of the content of the email body. In Pega Platform 8.6 and later, you use for this purpose the `pyEmailAcknowledgement` correspondence rule that takes into account the selected template and includes the security code tag. If your application uses a different rule, update this definition to match one of the built-in correspondence rules, for example, `EmailAckTemplate_Clear`.

Section `EmailFormat_<template_name>` - `@baseclass`

Defines a wrapper with HTML tags for your acknowledgment and reply outbound email templates. The system uses the HTML within this section as the template for the associated Correspondence rules. You typically define in this rule the heights and widths of the areas of

the email as well as the background colors. You also designate in the rule the header, body, and footer section of the email by using the *data-pegas-region* attribute applied to a div tag.

Section *EmailReplyBody_<template_name>* - *@baseclass*

Defines a standard section to include as the email body in your reply outbound email template. This section must include the code that iterates over the *EmailConversations* pagelist and builds up the HTML for the email reply in a thread. The system adds each email in the conversation thread in its own table so that the system can style the email.

Correspondence *EmailReplyTemplate_<template_name>* **Email** - *@baseclass*

Defines the email outbound reply template that uses the *EmailFormat_<template_name>* as the format and *EmailReplyBody_<template_name>* as the primary content body. You also specify the logo HTML in the header and copyright string in the footer in this rule.

Correspondence *EmailAckTemplate_<template_name>* **Email** - *Work-*

Defines the email outbound acknowledgment template that uses the *EmailFormat_<template_name>* as the format and the *pyEmailAckBody* section rule as the primary content body. You specify the logo HTML in the header and copyright string in the footer in this rule.



Note: The procedure shown below describes the steps required to create a new template using the recommended method of copying an existing version of the four outbound email templates. The steps cover copying the built-in Cobalt template to Green.

1. Switch to Dev Studio.
2. Create a new email format template with *EmailFormat_<template_name>* as a section rule:
 - a. In the search field, enter `_Cobalt`
 - b. In the search results, click **pyEmailFormat_Cobalt**.
 - c. Click the **HTML** tab, and in the **HTML source** field change the HTML to represent your company branding.
 - d. Click **Save as**, and then click **Save as**.
 - e. In the **Section Record Configuration** section, in the **Label** field, enter a title for the rule.
 - f. Click **Edit** next to the **Identifier** field, enter an identifier name in the following format: `EmailFormat_<name>`, and then click **OK**.

- g. Click **Create and open**.

The system saves the email format template as a section rule for the *@baseclass* class.

3. Create a new *EmailReplyBody_<template_name>* rule to include in your reply outbound email template:

- a. In the search field, enter `_Coba1t`
- b. In the search results, click **pyEmailReplyBody_Cobalt**.
- c. **Optional:** To change the reply body template information, click the **HTML** tab, and then in the **HTML source** field change the HTML so that it is consistent with your company branding.



CAUTION: This rule contains a combination of inline Java and HTML code. If you modify this code, you might break how the system renders the email reply body. You can only change the information within the *style* tags of the HTML elements for the rule.

- d. Click **Save as**, and then click **Save as**.
- e. In the **Section Record Configuration** section, in the **Label** field, enter a title for the rule.
- f. Click **Edit** next to the **Identifier** field, enter an identifier name in the following format: `EmailReplyBody_<name>`, and then click **OK**.
- g. Click **Create and open**.

The system saves the email format template as a section rule for the *@baseclass* class.

4. Create a new reply template *EmailReplyTemplate_<template_name>* as a correspondence rule:

- a. In the search field, enter `_Coba1t`
- b. In the search results, click **pyEmailReplyTemplate_Cobalt**.
- c. In the top-right corner of the page, click **Pick email theme**, and then in the **Available Templates** window, click the new email format template that you created in step 2.e (on page 86).
- d. Click the **Footer** section in the template, and then in the **Footer** section on the right, click the **HTML** tab.

- e. In the text area, edit the copyright information so that it references your company name.
- f. Click the **Primary content body** section in the template, and then in the **Primary content body** section on the right, click the **HTML** tab.
- g. In the text area, change the `pyEmailReplyBody_Cobalt` string to the name of the rule you created in step 3.f (on page 87).
- h. Click the **Header Logo** section in the template, in the **Header Logo** section on the right, click the **HTML** tab, and then in the text area, modify the HTML `img` tag attributes to reference an image for your company logo.
If you want to include your own image logos, host them at a publicly available URL rather than including them inline. Many email programs block inline images completely or send them as attachments, by default. As a result, broken email formatting or unnecessary attachments might occur in the system.
- i. Click **Save as**, and then click **Save as**.
- j. In the **Correspondence Record Configuration** section, in the **Label** field, enter a name for the new email reply template.
- k. Click **Edit** next to the **Identifier** field, enter an identifier name in the following format: `EmailReplyTemplate_<name>`, and then click **OK**.
- l. Click **Create and open**.

The system saves the email reply template as a correspondence rule for the `@baseclass` class.

5. Create a new acknowledgment template `EmailAckTemplate_<template_name>` as a correspondence rule:
 - a. In the search field, enter `_Cobalt`
 - b. In the search results, click **pyEmailAckTemplate_Cobalt**.
 - c. In the top-right corner of the page, click **Pick email theme**, and then in the **Available Templates** window, click the new email format template that you created in step 2.e (on page 86).
 - d. Click the **Header Logo** section in the template, in the **Header Logo** section on the right, click the **HTML** tab, and then in the text area, modify the HTML `img` tag attributes to reference an image for your company logo.

- e. Click the **Footer** section in the template, and then in the **Footer** section on the right, click the **HTML** tab.
- f. In the text area, edit the copyright information so that it references your company name.
- g. Click **Save as** and then **Save as**.
- h. In the **Correspondence Record Configuration** section, in the **Label** field, enter a name for the new email acknowledgment template.
- i. Click **Edit** next to the **Identifier** field, enter an identifier name in the following format: EmailAckTemplate_<name>, and then click **OK**.
- j. Click **Create and open**.

The system saves the email acknowledgment template as a correspondence rule for the *Work-* class.

6. Switch back to App Studio.

[Customizing the Email channel \(on page 76\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Enabling reports for the email bot

You can toggle the reports dashboard in the Email Manager portal and specify for the system to gather email triage case data for the built-in Pega Email Bot™ reports. After you enable reports in your email bot, the data in the reports can help you learn more about the email bot and improve the system, for example, by later adjusting the machine learning models to attain greater automation of the system.

The reports dashboard and the storing of triage cases are enabled in the system by default. If you set the *pyShowEmailAnalytics* when rule to *false*, the reports dashboard is not displayed and the system does not store triage actions in the *Index-TrackTriageActions* table.

To re-enable the reports dashboard and storage of triage cases, perform the following steps:

1. Switch to Dev Studio.
2. In Dev Studio, open the **pyShowEmailAnalytics** when rule.
3. In the when rule, click the **Advanced** tab.
4. Open the **pyEnableEmailAnalytics** rule, click the **Advanced** tab, and then set the value to *true*.
5. Click **Save**.
6. Switch back to App Studio.

[Viewing reports for the Email channel \(on page 72\)](#)

[Customizing the Email channel \(on page 76\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Adding the email bot reports dashboard to other portals

Customize the run-time environment of your application by also displaying the built-in email bot reports in the Case Manager or Case Worker portals. By default, the system displays the reports dashboard in the Email Manager portal. By adding the reports dashboard to other portals, you help customer system representatives (CSRs) gain access to valuable information about the level of automation and performance of the email bot. As a result, you can improve the artificial intelligence and text analysis of the email bot so that it responds to users more efficiently.

To insert the built-in email bot reports dashboard into a portal, you use the Live UI tool to add an embedded section to a user interface element, such as a table layout. The section with the reports dashboard that you insert into a portal is called *pyEmailAnalytics* and uses the *D_pyDataAnalytics* data page that is defined in the *Data-Channel-Analytics* class.



Note: You can also enable or disable the reports dashboard displayed in your portals. For more information, see [Enabling reports for the email bot \(on page 89\)](#).

1. Switch to Dev Studio.
2. Open a portal to which you want to add the reports dashboard by performing one of the following actions:
 - To open the Case Manager portal, in Dev Studio, click **Launch web interface**, and then click **Case Manager**.
 - To open the Case Worker portal, in Dev Studio, click **Launch web interface**, and then click **Case Worker**.
3. In the bottom-right of the page, click the **Toggle runtime toolbar** icon, and then click the **Toggle Live UI** icon.
For more information, see [Live UI \(on page 72\)](#).
4. Hover over and then click the user interface element to which you want to add the reports dashboard.
5. In the toolbar located to the right of the element's name, click the **Open in Dev Studio** icon.
6. In Dev Studio, in the displayed rule, select a reference element, and then in the toolbar at the top of the page, click the **Insert layout or row before selected** or the **Insert layout or row after selected** icon.

A new **Dynamic Layout** element is added before or after the reference element.

7. In the toolbar at the top of the page, click the **Structural** menu, and then drag the **Embedded section** into the new **Dynamic Layout** section.
8. In the **Section Include** window, in the **Page context** list, select **Use data page**.
9. In the **Class** field, enter Data-Channel-Analytics
10. In the **Data page** field, press the Down arrow key, and then select **D_pyDataAnalytics**.
11. In the **Section** list, select **By name**.
12. In the field to the right of the **Section** list, press the Down arrow key, and then select **pyEmailAnalytics**.
13. Click **Submit**.

The **Dynamic Layout** element displays the reports dashboard for the rule.

14. **Optional:** To move the reports dashboard into another place on the page, drag and drop the **Dynamic Layout** element before or after the other user interface elements.
15. Save your changes by performing one of the following actions:
 - To save the rule belonging to a locked ruleset to an isolated sandbox to test your changes, click **Private edit**.
 - To save the rule to a ruleset, click **Save as**.
16. Switch back to App Studio.

[Viewing reports for the Email channel \(on page 72\)](#)

[Customizing the Email channel \(on page 76\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Configuring how the email bot handles a reply to forwarded emails

By defining additional processing in an activity, you can customize how Pega Email Bot handles a reply received from a third-party stakeholder of a forwarded email, to more closely fit your needs and requirements. As a result, the system becomes more robust and responsive to the business processes that you define for your application. For example, you can configure your email bot to always send a copy of a received reply to a quality assurance assistant for further review.

To change how the email bot handles replies to forwarded emails, you override the *pyResponseToForwardExtension* activity rule for the *Work-Channel-Triage* class.

1. Switch to Dev Studio.
2. In the navigation pane of Dev Studio, click **Records**.
3. Expand the **Technical** section, and then click **Activity**.



The system displays all the activity rules in your application.

4. In the **Activity Name** column, click the **Click to Filter** icon.
5. In the dialog box, in the **Search Text** field, enter `pyResponseToForwardExtension`, and then click **Apply**.

The system displays the **pyResponseToForwardExtension** activity on the page.

6. Click the **pyResponseToForwardExtension** activity.
7. In the *pyResponseToForwardExtension* activity rule, configure which methods, step pages, and parameters the system invokes to perform customization of a reply to forwarded emails.
For more information, see *Creating an activity (on page 76)*.
8. Save your changes by performing one of the following actions:
 - To save the rule that belongs to a locked ruleset to an isolated sandbox so that you can test your changes, click **Private edit**.

For more information, see *Performing a private edit (on page 76)*.
 - To save the rule to a ruleset, click **Save as**.
9. Switch back to App Studio.

[Configuring the email bot to display a user name in emails \(on page 92\)](#)

[Customizing the Email channel \(on page 76\)](#)

Configuring the email bot to display a user name in emails

Modify your system to display a user name in a specific format by default in emails that customer service representatives (CSRs) process in the Email Manager and Case Manager portals for your application. As a result, you fine tune the user experience of these portals to the requirements of your organization, making the system both easier to use and more user friendly. For example, you can configure the system to display the user's first and last name, separated by a period, in a reply.

To customize your system to display a user name in a specific format in emails, you configure the *pyGetDisplayNameForEmailAddress* data transform rule for the *Work-Channel-Triage* class. By default, the system displays the user's name in the reply in the following format: `LastName.FirstName`.

1. Switch to Dev Studio.
2. In the navigation pane of Dev Studio, click **App**, and then search for the *Work-Channel-Triage* class.
3. Expand the **Data Model > Data Transform** section for the *Work-Channel-Triage* class, and then click **pyGetDisplayNameForEmailAddress**.
4. In the *pyGetDisplayNameForEmailAddress* data transform rule, configure the actions and target parameters to format the display name or email address in emails.

For more information, see [Configuring a data transform \(on page 91\)](#).

5. Save your changes by performing one of the following actions:

- To save the rule that belongs to a locked ruleset to an isolated sandbox so that you can test your changes, click **Private edit**.

For more information, see [Performing a private edit \(on page 91\)](#).

- To save the rule to a ruleset, click **Save as**.

6. Switch back to App Studio.

[Configuring how the email bot handles a reply to forwarded emails \(on page 91\)](#)

[Customizing the Email channel \(on page 76\)](#)

Overriding the default language for an Email channel

If you plan to support multiple languages in your application or one language other than English, you can override the default language of a new Email channel. Overriding the default language ensures that Pega Email Bot™ processes emails from users in the correct language, performs text analysis and provides automatic responses. By default, the system processes emails in English when you create an Email channel. For example, if an organization is interested in creating an email bot so that the system processes emails in French, you can specify French as the default language. Otherwise, if the organization does not change this setting, the email bot processes emails in English, by default.

1. Switch to Dev Studio.
2. In the navigation pane of Dev Studio, click **App**, and then search for the *Data-Channel-* class.
3. Expand the **Data Model > Data Transform** section for the *Data-Channel-* class, and then click **pyGetDefaultLanguage**.
4. On the **Definition** tab in the *GetDefaultLanguage* data transform rule, in the **Source** column, enter a language in double quotation marks.

The following figure shows the *GetDefaultLanguage* data transform rule with the default language overridden to German:

Definition Parameters Pages & Classes Test cases History				
	Action	Target	Relation	Source
• 1	Comment	For upgrade scenarios where no default language is set, language set in this parameter is taken as default language		
• 2	Set	Param.DefaultLanguage	equal to	"German"
+ Collapse All Expand All ☒ Call superclass data transform				

The *Param.DefaultLanguage* target set to "German" in the *GetDefaultLanguage* data transform

5. Click **Save**.

6. Switch back to App Studio.



[Creating an Email channel \(on page 26\)](#)

Enabling training the model based on email attachments

Apart from the content of an email body, you can optionally train the text analytics model for your Pega Email Bot™ with the information in email attachments. As a result, you make the training process in your system more consistent and you improve the accuracy of natural language processing (NLP) analysis. For example, customers interact with your email bot to apply for a bank loan, during which they attach files describing their financial credit history. You can enhance the training the model in the system by using training data in file attachments that contain sample credit histories. As a result, the email bot can then improve the detection of correct entities and topics from customers in both the body of their emails and in their file attachments.

By default, you train the text analytics model in your system by using training data from email bodies. By enabling the training of model based on email attachments, you can also use training data from text files, and image-based files if you are using an external optical character recognition (OCR) component in your system.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the Email channel, click the **Behavior** tab.
5. In the **Text Analyzer** section, select the **Record training data** checkbox.
6. **Optional:** To use an advanced text analyzer configuration, perform the following steps:
 - a. In the **Text Analyzer** section, select the **Use advanced configuration** checkbox.
 - b. Click **Add text analyzer**, or if the iNLP text analyzer type is already displayed in the list, click the **Switch to edit mode** icon in the iNLP row.
 - c. In the **Text analyzer configuration** window, in the **Text Analyzer type** list, select **iNLP**.
 - d. Select the **Enable model training** check box.
 - e. Click **Submit**.
7. Click **Save**.
8. Switch to Dev Studio.
9. In the navigation pane of Dev Studio, click **App**, and then search for the *Work-Channel-Triage-Email* class.
10. Expand the **Data Model > Data Transform** section for the *Work-Channel-Triage-Email* class, and then click **pyDefault**.
11. In the *pyDefault* data transform rule, in the table row for the **pyEnableAttachmentTraining** target, in the **Source** column, enter: `true`

12. **Optional:** To increase the limit on the amount of text extracted from file attachments in the system, in the table row for the **pyExtractedTextLimit** target, in the **Source** column, enter the maximum number of bytes to extract as text from file attachments.
13. Save your changes by performing one of the following actions:
 - To save the rule that belongs to a locked ruleset to an isolated sandbox so that you can test your changes, click **Private edit**.

For more information, see [Performing a private edit \(on page 76\)](#).
 - To save the rule to a ruleset, click **Save as**.
14. Switch back to App Studio.

[Customizing the Email channel \(on page 76\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Troubleshooting the email bot

Increase the availability and responsiveness of Pega Email Bot™ so that users interact with the application without a delay and the system escalates reported problems to the correct operator or work queue.

You can address email bot performance issues and test how the system reacts to emails. By resolving problems that are related to the email account, channel administrators ensure that the system can send and receive emails. Similarly, channel developers verify that routing conditions that are defined for intelligent routing generate the expected results.

Testing all the routing conditions for the email bot ensures that the system can intelligently respond to requests, for example, by starting a business case in the application and sending an automatic reply.



Note: To learn more about troubleshooting natural language processing (NLP) issues in your email bots, see also [Troubleshooting NLP](#).

1. Resolve issues with the email account for the Email channel.
For more information, see [Troubleshooting email accounts in the Email channel \(on page 96\)](#).
2. Check the defined routing conditions for the Email channel.
For more information, see [Troubleshooting email routing conditions \(on page 97\)](#).
3. Fix issues related to performance for the Email channel.
For more information, see [Troubleshooting other email bot issues \(on page 98\)](#).

[Conversational channel user roles \(on page 19\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

Troubleshooting email accounts in the Email channel

Ensure that Pega Email Bot™ can react to emails at any time so that user issues are quickly addressed in the application, by verifying that the email settings for the channel always work as expected. As a result, users can interact with the email bot to report an issue or ask for more information without any delay. For example, when the email bot cannot receive emails, a channel administrator can quickly troubleshoot the issue in a variety of ways by performing such actions as restarting the email listener or minimizing throttling of the system.

1. Ensure that a dedicated instance of the email account and email listener is created in Pega Platform. You cannot use an email account and email listener for an Email channel if other services in the application already use the listener.
2. Ensure that the dedicated email listener completes these tasks:

- a. Starts and runs continuously in the application.
- b. Processes emails and sends autoreply messages.
- c. Processes an email by marking the content in its thread as read.
- d. Processes text formatting, images, and attachments in emails correctly, including large files.
- e. Does not accumulate too many emails in the inbox causing throttling and other performance issues.

Throttling occurs when the email listener drops the connection or rejects additional connections if the number of incoming email messages surpasses a certain threshold. For more information, see [Throttling during email processing \(on page 95\)](#).

To learn how to resolve email listener issues and optimize its performance, see [Troubleshooting email listeners \(on page 95\)](#) and [Best practices for optimizing email listener performance \(on page 95\)](#).

3. If issues persist in the system, restart the email listener. For more information, see [Restarting an email listener \(on page 95\)](#).

[Creating an Email channel \(on page 26\)](#)

[Troubleshooting the email bot \(on page 95\)](#)



Troubleshooting email routing conditions

During testing of the routing conditions that you define for intelligent routing, Pega Email Bot™ might not always generate the correct outcome. To solve this issue, there are several guidelines that you can follow while testing your routing conditions in the system. Troubleshooting issues and verifying all of the expected actions that trigger during intelligent routing ensures that the email bot works as expected and improves the availability of the system in a production environment.

1. Check the email bot settings for natural language processing (NLP):
 - a. Configure one or more advanced text analyzers in the system, for example, the iNLP text analyzer. For more information, see [Adding a text analyzer for an email bot \(on page 36\)](#).
 - b. Define the topics (the subject matter of emails), that the system will detect during intelligent routing. For more information, see [Defining topics for text analysis for an email bot \(on page 39\)](#).
 - c. If you want the email bot to also analyze attachments during intelligent routing, enable email attachments analysis in the system. For more information, see [Enabling email attachments analysis during email triage \(on page 42\)](#).
 - d. If you want the email bot to support multiple languages, enable automatic language detection during intelligent routing. For more information, see [Enabling automatic language detection for text analysis \(on page 43\)](#).
2. Verify the expected email bot behavior for each routing condition:
 - a. Add all suggested cases to start in the system as a result of intelligent routing. For more information, see [Adding suggested cases for an Email channel \(on page 30\)](#).
 - b. If applicable, ensure that the system can route emails to the correct operator or work queue.
3. Continue to train the text analytics model for the email bot:
 - a. Create enough instances of sample training data to trigger the action for each defined routing condition. For more information, see [Creating training data manually for an email bot \(on page 50\)](#).
 - b. Modify the training data to detect the correct topic, sentiment, language, and entities that are used in the routing conditions. For more information, see [Correcting training data in an email bot \(on page 51\)](#).
 - c. After you rebuild the model, test the routing conditions once again until they trigger the correct action. Repeat this process until you have verified the correct outcomes for each of the routing conditions. For more information, see [Testing email routing conditions \(on page 47\)](#).

[Configuring intelligent email routing \(on page 44\)](#)

[Troubleshooting the email bot \(on page 95\)](#)



Troubleshooting other email bot issues

Fix performance and other issues in the Pega Email Bot™ to ensure that the system works as expected and that customers can interact with the email bot in a timely manner.

You might encounter the following issues:

- The system generates PEGA0039 alerts and opening an email triage or interaction case is slower than expected (*on page 98*)
- You cannot access an email correspondence rule on the Behavior tab in the Email channel (*on page 98*)

The system generates PEGA0039 alerts and opening an email triage or interaction case is slower than expected

Duplicate data in email triage cases might cause a large clipboard page size in the system. As a result, Pega Platform™ is slower than expected when you open an email triage or interaction case, and the system also generates PEGA0039 alerts.

The PEGA0039 alert provides a warning that the memory performance of your system might be affected when Pega Platform expands the BLOB onto a clipboard page. For more information, see [PEGA0039 alert: BLOB size read from the database exceeded threshold](#).

To solve this issue, update to Pega Platform 8.7.1 or higher. This version introduces performance improvements, which help reduce the clipboard page size by not storing email body, attachments, and NLP outcome data in this page.

1. Update to Pega Platform 8.7.1 or higher.
2. For backward compatibility purposes, replace any email body references in the clipboard page with references to one of the new data pages.

For more information, see [Updating the email bot to the 8.7.1 version \(on page 99\)](#).

You cannot access an email correspondence rule on the Behavior tab in the Email channel

You create an email correspondence rule manually in Dev Studio. When you try to access the new correspondence rule in the Email channel on the **Behavior** tab, you cannot select this new suggested reply template in the **Add suggested reply** list.

To solve this issue, always create a correspondence rule in an open ruleset for your application. Ensure you create the new correspondence rule while you are configuring the Email channel so that the system creates this template in an open ruleset.



1. In the Email channel, click the **Behavior** tab.
2. Click **Add suggested reply**.
3. In the **Add suggested reply** window, click **Create new**.
4. In the **Name** field, enter a label for the email response template.
5. In the text area, enter the default content for the email response template.
6. Click **Submit**.

For more information, see [Defining suggested responses for an Email channel \(on page 33\)](#).

[Creating an Email channel \(on page 26\)](#)

[Troubleshooting the email bot \(on page 95\)](#)

Updating the email bot to the 8.7.1 version

Take advantage of Pega Email Bot™ performance improvements that are available in Pega Platform™ version 8.7.1. As a result, you enhance how the system references email triage information, helping customers to interact with the email bot in a timely manner.

Pega Platform version 8.7.1 introduces performance improvements that help reduce the clipboard page size by not storing email body, attachments, and NLP outcome data in this page. The following data pages are now available so that you can obtain the relevant email body for an email triage or interaction case that was previously referenced in the clipboard page:

- *D_pxFirstInboundEmail[ETCaseId: "ET-100"].pyBody*
- *D_pxLatestInboundEmail[ETCaseId: "ET-100"].pyBody*
- *D_pxAllInboundEmails[ETCaseId: "ET-100"].pxResults(1).pyBody*

For compatibility purposes, if you update to Pega Platform 8.7.1 or higher:

1. Identify rules in your Pega Platform application that refer to properties where the email body for an email triage case is referenced in the *pyWorkPage* clipboard page.
2. Replace the references in these rules with the email body references by using one of the three new data pages.

[Creating an Email channel \(on page 26\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

Updating to the threading mechanism available in the 8.6 version

To help track emails from customers and other stakeholders in separate threads, starting from Pega Platform™ version 8.6, the system includes a more secure threading mechanism. If you update to Pega Platform version 8.6 and in the previous version of the system you configured an Email channel, perform additional configuration steps in order to use the built-in secure threading mechanism with your email bot.



To ensure that you use the new secure threading mechanism for your email bot, perform the following actions:

- Update the service email rule for your email bot so that the system uses an initial acknowledgment email.
- Update the *pySetEmailBodyReplyTemplate* data transform rule so that the system uses an email reply template.



Note: Base the contents of the initial acknowledgment email and email reply template in your system on one of the built-in email outbound templates, which includes an embedded security code tag. For more information, see [Creating outbound email templates \(on page 85\)](#).

1. Switch to Dev Studio.
2. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. In the **Email handling** section, in the **Email accounts** list, write down the name of the email listener rule that the system uses for the Email channel.
5. In the header of Dev Studio, click **Configure > Integration > Email > Email Listeners**.

The system displays a list of email listener rules for your application.

6. In the **Listener name** column, click the name of the email listener that the system uses for your Email channel. This is the name that you made a note of in step 4 ([on page 100](#)).
7. In the email listener rule, in the **Listener properties** section, edit the rule displayed in the **Service method (text/XML only)** field.
8. In the service email rule, click the **Response** tab.
9. Expand the first **Message contents** section, and then in the **Message data** section for the email body, in the **Map from Key** field, edit the email correspondence rule for your email bot.
In Pega Platform version 8.6 and later, by default the system uses the *pyEmailAcknowledgement* correspondence rule for this purpose.
10. If your application uses a different email correspondence rule from *pyEmailAcknowledgement*, in the email correspondence rule, on the **Corr** tab, in the text field, reference one of the built-in acknowledgment templates: *pyEmailAckTemplate_Clear*, *pyEmailAckTemplate_Cobalt*, *pyEmailAckTemplate_Classic*, or a custom template that also uses the *pyEmailFormat_<template_name>* rule as the wrapper with the HTML tags including the embedded security tag.

By referencing an acknowledgment template that includes the *pyEmailFormat_<template_name>* rule, you ensure that you use the secure built-in threading mechanism. The system includes the following tag in the HTML source for this rule:

```
<pega:include name="pyEmailThreadIdentification" type="Rule-HTML-Paragraph"></pega:include>
```

11. Save your changes by performing one of the following actions:

- To save the rule that belongs to a locked ruleset to an isolated sandbox so that you can test your changes, click **Private edit**.

For more information, see [Performing a private edit \(on page 85\)](#).

- To save the rule to a ruleset, click **Save as**.

12. In the navigation pane of Dev Studio, click **App**, and then search for the *Work-Channel-Triage* class.

13. Expand the **Data Model > Data Transform** section for the *Work-Channel-Triage* class, and then click **pySetEmailBodyReplyTemplate**.

14. In the *pySetEmailBodyReplyTemplate* data transform rule, set the parameter reply template target to one of the new built-in email reply templates: *pyEmailReplyTemplate_Clear*, *pyEmailReplyTemplate_Cobalt*, or *pyEmailReplyTemplate_Classic*.

You can also use your own custom email reply template, provided that it is based on one of the three built-in templates. For more information, see [Creating outbound email templates \(on page 85\)](#).

15. Save your changes by performing one of the following actions:

- To save the rule that belongs to a locked ruleset to an isolated sandbox so that you can test your changes, click **Private edit**.

For more information, see [Performing a private edit \(on page 85\)](#).

- To save the rule to a ruleset, click **Save as**.

16. Switch back to App Studio.

[Email triage process \(on page 57\)](#)

[Creating an Email channel \(on page 26\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

Updating the email bot to the 8.5 version

Improve Pega Email Bot™ performance and ensure that the email bot is more reliable and working as expected by carrying out update and maintenance procedures. When you take advantage of the latest features of the system, you maintain the better quality and health of the product, which enables users to interact with your application more efficiently.

Channel administrators carry out tasks related to updates and maintenance of the email bot when the system is in a development, QA, or production environment.

- Migrate existing database table contents to new tables for the 8.5 version to improve the performance of the email bot.

For more information, see [Migrating triage cases to new tables \(on page 102\)](#).

- If your system is installed in Pega Cloud Services, schedule maintenance of the system by periodically archiving triage cases.

For more information, see [Archiving resolved emails for an email bot \(Pega Cloud Services\) \(on page 104\)](#).

[Conversational channel user roles \(on page 19\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

Migrating triage cases to new tables

Starting from Pega Platform 8.5, the system uses new tables to store complete information about triage cases for Pega Email Bot™. To ensure that the email bot works as expected and to improve its overall performance, Pega Platform migrates all of your resolved and open triage cases to a new table during updates to version 8.5 or later. Once you begin the update process for your installation of Pega Platform, the system performs the migration automatically in the background using a dedicated job scheduler. If you expose additional properties as optimized columns in the old tables, before you start the update, extend the *pyPopulateColumnNamesExtension* data transform with this information.

If you are updating to Pega Platform 8.5, during the update process the system automatically migrates existing email triage cases from the *pc_work* table to the new *pc_work_triage* table. As a result, you can later use the email bot to view built-in reports using the information from your triage cases. For more information, see [Viewing reports for the Email channel \(on page 72\)](#) and [Database schema for email bot tables \(on page 105\)](#).

Updating your version of Pega Platform occurs either as an out-of-place update or an in-place update. You perform an out-of-place update on Pega Platform running in Pega Cloud Services. This type of update has almost no downtime. You perform an in-place update for an on-premises or a non-Pega Cloud Services version of Pega Platform. This type of update requires some downtime during the process. For example, a system with 1 million triage cases stored in database tables completes the migration process in the background in approximately 4 hours.



Migration job scheduler

To migrate all resolved and open triage cases to the new *pc_work_triage* table, the system runs the *pyMigrateInteractionCases* job scheduler defined in the *Work-Channel-Triage* class in the background, during your system update. The job scheduler migrates unresolved triage cases first, and once complete, the system migrates the resolved triage cases. The job scheduler migrates 5,000 triage cases to the new table at a time, sleeps for one minute, and then repeats the process until all the data is successfully migrated. In addition, if the job scheduler unexpectedly goes down when it is running for any reason, and some triage cases are not migrated, the job scheduler picks up these triage cases for migration in its next run cycle.

Migration of exposed columns

If you expose additional properties as database columns in the *pc_work* table and optimize the properties for reporting, perform an initial configuration step before you begin updating your system. For this purpose, you extend the *pyPopulateColumnNamesExtension* data transform. If you expose additional custom columns in the *pc_work* table, you migrate your exposed columns by first defining them in the *pyPopulateColumnNamesExtension* data transform, and then beginning the update of your system. During the migration process, which runs in the background, the *pyMigrateInteractionCases* job scheduler optimizes the exposed columns based on what you define in this data transform.

For example, if you previously customize three columns in the *pc_work* table, the *pyPopulateColumnNamesExtension* data transform rule to which you add the three custom exposed columns for optimization in the new table appears as in the following figure:

Home pyPopulateCol... TC_PopulateCo...

Edit Data Transform: **pyPopulateColumnNamesExtension** [Available]
 CL: Work-Channel-Triage-Email ID: pyPopulateColumnNamesExtension RS: Cholcapp:01-01-01

Definition Parameters Pages & Classes Test cases Specifications History

	Action	Target	Relation	Source
1	Comment	Use this DT when there are customised columns in pc_work table. Let us assume that CountryName is customised column in pc_work then try		
2	Set	OptimizeColumnPages.pxResults(<APPEND>).pyLabel	equal to	"ExposedCloumn1" Select values +
3	Set	OptimizeColumnPages.pxResults(<APPEND>).pyLabel	equal to	"ExposedCloumn2" Select values +
4	Set	OptimizeColumnPages.pxResults(<APPEND>).pyLabel	equal to	"ExposedCloumn3" Select values +

+ Collapse All Expand All

☒ Call superclass data transform

Sample of the *pyPopulateColumnNamesExtension* data transform rule

Note: You can also perform migration of exposed columns using the

pyPopulateColumnNamesExtension data transform after the update process completes, provided that the *pyMigrateInteractionCases* job scheduler is still running in the background.

[Updating the email bot to the 8.5 version \(on page 101\)](#)

[Customizing the Email channel \(on page 76\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Archiving resolved emails for an email bot (Pega Cloud Services)

Archive resolved emails for triage cases in Pega Email Bot™ that are older than a specified number of days in Pega Cloud Services installations of Pega Platform. Since the email triage data in tables can grow quickly for an email bot that is available in Pega Cloud Services, archiving resolved emails improves the overall performance of your system by reducing primary storage consumption and cost. Emails from email triage cases that are resolved and archived are no longer actively used by the email bot, and are placed in secondary storage.

When the email bot receives an email that is related to an archived, resolved email triage case, the system then also displays the archived triage case in the context of the new triage case for reference purposes, for example, in the Email Manager portal.

You use the *Data-Retention-Policy* rule for the *Work-Channel-Triage-Email* class to control which cases your application moves from primary storage to secondary storage. The system runs the archive process for this policy in the background using system agents and job schedulers. For more information, see Archiving and expunging case data (on page).

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Email channel.
4. On the Email channel configuration page, click the **Behavior** tab.
5. In the **Options** section, select the **Archive emails** checkbox.
6. In the **Archive resolved emails older than** field, enter the number of days after which the system will archive resolved triage cases.
By default, the system archives emails that are older than 365 days.
7. Click **Save**.

[Updating the email bot to the 8.5 version \(on page 101\)](#)

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Database schema for email bot tables

In Pega Platform 8.5 and later, Pega Email Bot™ uses two new tables that list complete information about email triage cases and captured actions performed for triage cases in the system. The system uses the information stored in these tables to display the built-in reports for the email bot. As a result, with the information from the reports, you can adjust the machine learning models to attain greater automation in the system. You can also create custom reports for the email bot, if you extend the columns in the two new tables by running data transforms.

pc_work_triage table

The *pc_work_triage* table for the *Work-Channel-Triage* class stores complete information about email triage cases for the email bot. In Pega Platform 8.4 and earlier, the system stores email triage cases in the *pc_work* table. When you update to Pega Platform 8.5 and later, the system automatically performs migration of resolved and open triage cases for the email bot from the old *pc_work* table to the new *pc_work_triage* table. If you expose additional properties as columns in the old table, before you start the update process extend the *pyPopulateColumnNamesExtension* data transform with this information. For more information, see [Migrating triage cases to new tables \(on page 102\)](#).

The *pc_work_triage* table consists of the following columns:

Column	Description	Data type	Length
<i>pxcommitdatetime</i>	The time and date when the case was committed.	Timestamp	
<i>pxsavedatetime</i>	The time and date when the case was last saved.	Timestamp	
<i>pxcoverinskey</i>	The cover ins key.	Character	var (255)
<i>pxcoveredcount</i>	The covered count.	Numeric	num (18,0)
<i>pxcoveredcountopen</i>	The open covered count.	Numeric	num (18,0)
<i>pxcoveredcountunsatisfied</i>	The unsatisfied covered count.	Numeric	num (18,0)
<i>pxcreatedatetime</i>	The time and date when the case was created.	Timestamp	
<i>pxcreateopname</i>	The name of the operator that created the case.	Character	var (128)
<i>pxcreateoperator</i>	The operator that created the case.	Character	var (128)
<i>pxcreatesystemid</i>	The identifier of the system that created the case.	Character	var (32)
<i>pxflowcount</i>	The flow count.	Numeric	num (18,0)

Column	Description	Data type	Length
<i>pxinsname</i>	The ins name.	Character	var (128)
<i>pxobjclass</i>	The name of the class object.	Character	var (96)
<i>pxupdatedatetime</i>	The time and date when the case was updated.	Timestamp	
<i>pxupdateopname</i>	The name of the operator that updated the case.	Character	var (128)
<i>pxupdateoperator</i>	The operator who updated the case.	Character	var (128)
<i>pxupdatesystemid</i>	The identifier of the system that performed an update.	Character	var (32)
<i>pyagefromdate</i>	The from date.	Timestamp	
<i>pydescription</i>	The description of the case type.	Character	var (255)
<i>pyeffortactual</i>	The actual effort.	Numeric	num (18,0)
<i>pyeffortestimate</i>	The estimated effort.	Numeric	num (18,0)
<i>pyeffortestimatetime-stamp</i>	The estimated effort time and date.	Timestamp	
<i>pyelapsedpastdeadline</i>	The elapsed time when the case is past the SLA deadline.	Numeric	num (18,0)
<i>pyelapsedpastgoal</i>	The elapsed time when the case is past the SLA goal.	Numeric	num (18,0)
<i>pyelapsedstatusnew</i>	The elapsed time when the case status is new.	Numeric	num (18,0)
<i>pyelapsedstatusopen</i>	The elapsed time when the case status is open.	Numeric	num (18,0)
<i>pyelapsedstatuspending</i>	The elapsed time when the case status is pending.	Numeric	num (18,0)
<i>pyeventid</i>	The identifier of the event for the case.	Character	var (32)
<i>pyid</i>	The identifier of the case.	Character	var (32)
<i>pylabel</i>	The label of the case.	Character	var (64)
<i>pyorigdivision</i>	The division of the user who created the case.	Character	var (32)
<i>pyorigorg</i>	The organization of the user who created the case.	Character	var (32)
<i>pyorigorgunit</i>	The organizational unit of the user who created the case.	Character	var (32)

Column	Description	Data type	Length
<i>pyoriguserdivision</i>	The division of the user who created the case.	Character	var (32)
<i>pyoriguserid</i>	The identifier of the user who created the case.	Character	var (128)
<i>pyoriguserworkgroup</i>	The work group of the user who created the case.	Character	var (64)
<i>pyownerdivision</i>	The division of the user who is the owner.	Character	var (32)
<i>pyownerorg</i>	The organization of the user who is the owner.	Character	var (32)
<i>pyownerorgunit</i>	The organizational unit of the user who is the owner.	Character	var (32)
<i>pyreopencount</i>	The number of open cases.	Numeric	num (18,0)
<i>pyreopentimestamp</i>	The time and date when the case was opened.	Timestamp	
<i>pyresolutioncost</i>	The cost of the resolution.	Numeric	num (18,0)
<i>pyresolveddivision</i>	The division of the user who resolved the case.	Character	var (32)
<i>pyresolvedorg</i>	The organization of the user who resolved the case.	Character	var (32)
<i>pyresolvedorgunit</i>	The organizational unit of the user who resolved the case.	Character	var (32)
<i>pyresolvedtime</i>	The time of resolution.	Character	var (32)
<i>pyresolvedtimestamp</i>	The time and date of the resolution.	Timestamp	
<i>pyresolveduserid</i>	The identifier of the user who resolved the case.	Character	var (128)
<i>pyresolveduserwork-group</i>	The user group of the user who resolved the case.	Character	var (64)
<i>pysladeadline</i>	The time and date of the service-level agreement (SLA) deadline.	Timestamp	
<i>pyslagoal</i>	The time and date of the service-level agreement (SLA) goal.	Timestamp	
<i>pyslaname</i>	The name of the service-level agreement (SLA).	Character	var (32)
<i>pystatuswork</i>	The current status of the work item.	Character	var (32)
<i>pyurgencyadjustment</i>	The urgency adjustment.	Numeric	numeric (18,0)

Column	Description	Data type	Length
<i>pxcurrentstagelabel</i>	The name of the current stage of the case type.	Character	var (128)
<i>pxcurrentstage</i>	The current stage of the case type.	Character	var (32)
<i>pzinskey</i>	The ins key.	Character	var (255)
<i>pxapplication</i>	The application identifier.	Character	var (64)
<i>pxapplicationversion</i>	The version of the application.	Character	var (32)
<i>pzpvstream</i>	The PVS stream.	Byte	
<i>pymodificationdatetime</i>	The time and date of last modification.	Timestamp	
<i>pyapplication</i>	The name of the application.	Character	var (64)
<i>pyowneruserid</i>	The identifier of the owner.	Character	var (128)
<i>pyslaaction</i>	The service-level agreement (SLA) action.	Timestamp	
<i>pxstagelabel</i>	The name of the stage for the case type.	Character	var (128)
<i>pxstage</i>	The stage for the case type.	Character	var (32)
<i>pyCurrentSentiment</i>	The sentiment of the latest email received from a user.	Character	var (32)
<i>pyBaseLanguage</i>	The detected language of the email received from a user.	Character	var (64)
<i>pyAccountId</i>	The account identifier for the Email channel.	Character	var (64)
<i>pyPrimaryTopic</i>	The detected topic, the subject matter of the received email.	Character	var (64)
<i>pySenderName</i>	The name of the user who sent the email.	Character	var (256)
<i>pySenderEmail</i>	The email address of the user who sent the email.	Character	var (128)
<i>pyChannelClass</i>	The name of the class for the Email channel.	Character	var (128)
<i>pyChannelID</i>	The identifier of the Email channel.	Character	var (256)
<i>pyCustomerInteraction-Sequence</i>	The customer interaction sequence.	Numeric	numeric (18,0)
<i>pzlInteractionCaseVer</i>	The Pega Platform version.	Character	var (32)

pr_index_tracktriageactions table

The *pr_index_tracktriageactions* table for the *Index-TrackTriageActions* class stores information about captured actions that are performed on a triage case in the system. The system uses the data stored in columns for this table to display the built-in or custom reports for your email bot. The table includes several essential columns that you can use to generate custom reports. You generate a custom report by first configuring the *pyAddTriageActionExtension* data transform to add custom columns to this table, and then running the data transform when you want to generate a custom report.



Note: You can also extend the *pr_index_tracktriageactions* table to generate reports for Pega Intelligent Virtual Assistants (IVAs), for example, IVA for Legacy Webchat.

The *pr_index_tracktriageactions* table consists of the following columns:

Column	Description	Data type	Length
<i>pxinsindexedkey</i>	The ins index key.	Character	var (255)
<i>pxindexcount</i>	The count of the index.	Numeric	numeric (18,0)
<i>pxindexpurpose</i>	The purpose index.	Character	var (32)
<i>pxcommitdatetime</i>	The date and time when the commit occurred.	Timestamp	
<i>pxsavedatetime</i>	The saved date and time.	Timestamp	
<i>pxinsname</i>	The ins name.	Character	var (128)
<i>pxobjclass</i>	The object class.	Character	var (128)
<i>pyActionType</i>	The triage action type: Case, Reply, RouteOp, RouteWQ.	Character	var (64)
<i>pyActionMode</i>	The triage action mode: Manual, Automatic.	Character	var (32)
<i>pyTarget</i>	The target name for the operator ID or work queue ID that also includes the case type class name.	Character	var (128)
<i>pyTargetID</i>	The target ID with case ID and pulse ID.	Character	var (64)
<i>pyThreadCount</i>	The thread count for email.	Decimal	numeric (18,0)
<i>pyTriageActionSequence</i>	The triage action sequence number.	Decimal	var (18,0)

Column	Description	Data type	Length
<i>pyAccountID</i>	The account identifier for the Email channel.	Character	var (64)
<i>pyApplication</i>	The name of the application.	Character	var (64)
<i>pyApplicationVersion</i>	The version of the application.	Character	var (32)
<i>pyBaseLanguage</i>	The detected language of the email received from a user.	Character	var (64)
<i>pyChannelClass</i>	The name of the class for the Email channel.	Character	var (128)
<i>pyChannelID</i>	The identifier of the Email channel.	Character	var (64)
<i>pyPrimaryTopic</i>	The detected topic, the subject matter of the received email.	Character	var (64)
<i>pySentiment</i>	The detected sentiment for the received email which is the same for all actions performed in the email thread.	Character	var (32)
<i>pyWorkCreateDateTime</i>	The date and time when the work item was created.	Timestamp	
<i>pyWorkObjClass</i>	The class for the work item.	Character	var (128)
<i>pyWorkObjectStatus</i>	The work item status, for example, resolved-completed or open.	Character	var (32)

[Migrating triage cases to new tables \(on page 102\)](#)

[Updating the email bot to the 8.5 version \(on page 101\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Creating an Email channel from an existing channel

Decrease the time a channel developer needs to build Pega Email Bot™ by creating an Email channel based on a verified configuration in another Email channel. For example, you have a tested email bot in a production environment that intelligently responds to customer requests, starts business cases, and interacts with your Pega Platform™ application. Consequently, you can reuse the Email channel settings in a new channel.

When you create an Email channel based on a configuration in another channel, the system copies the following information from the existing channel:

- Suggested cases
- Suggested replies



- Configured text analyzers
- Intelligent routing settings

The system does not copy the following information from the existing channel:

- Configured email accounts
- Reference operator
- Supported languages
- Selected outbound email template

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. Copy existing Email channel settings to a new channel:

- a. In the **Current channel interfaces** section, click the icon that represents the existing Email channel that has the settings that you want to copy to the new channel.

The system displays the configuration for the Email channel.

- b. In the upper right of the page, click **Actions > Copy channel interface**.
- c. In the **Copy channel interface** window, click **Copy**.

The system displays a new Email channel.

4. Configure the new Email channel:
 - a. In the **Details** section, in the **Email channel name** field in the **Details** section, enter a name for your Email channel.
 - b. **Optional:** To explain the purpose of your email bot, in the **Description** field, enter a short description for the email bot.
 - c. In the **Email handling** section, click **Add email account**, and then select an email account. Set up at least one email account for the Email channel so that users know which email address to use for sending their requests to the email bot. If you set up more than one email account, users can send emails to the email bot from any of the email addresses for the defined email accounts.
 - d. **Optional:** To enhance how the system presents information in emails and triage cases by enabling rich HTML text, in the **Additional settings** section, select the **Allow rich text (HTML) in emails** checkbox.

- e. In the **Reference operator** list, select a reference operator for your email bot.
You select a reference operator so that the system assigns the email triage case to a unique Pega Platform operator when an exception occurs. A reference operator is a Pega Platform user on whose behalf the system creates service mails in the system.
 - f. In the **Languages** list, select the languages that the email bot uses to interact with users.
 - g. In the **Outbound email template** list, select a template for the system to use for outbound emails.
5. Click **Save**.

[Implementing an email bot for an Email channel \(on page 25\)](#)

[Pega Email Bot overview \(on page 5\)](#)

Implementing an Intelligent Virtual Assistant for a conversational channel

Provide users with the ability to interact with your application by getting answers to simple questions and completing complex work, without having to engage with or wait for a human agent. Pega Intelligent Virtual Assistant™ (IVA), which you design in the system, interacts with users in a simple question and answer form by sending text messages, and by using artificial intelligence, natural language processing (NLP), and your existing Pega Platform™ case processing. The convenience of this chat functionality is available across multiple messaging platforms and ensures that you instantly provide essential business services to users in a consistent way. For example, users can quickly request information about their account or even update the account directly by interacting with your chatbot in Apple Messages for Business, Facebook Messenger, WhatsApp, and other messaging platforms.

IVA for Digital Messaging includes a built-in preview console. You use the preview console to test the chatbot during the design phase, before moving the bot to a production environment. You test and configure your chatbot on the same page. You can also train machine learning models to recognize the way in which your customers communicate with you, without any data science expertise or a single line of code.

Note:



- The process of creating a Pega Platform application, including the design and development of an IVA channel for a conversational channel, requires open rulesets. However, after moving your IVA to a production environment, you can lock the rulesets. Only when you



need to build models further or change the IVA channel configuration in a production environment, must you then unlock the rulesets.

- This release introduces new built-in data types, *Conversation Event* and *Conversation Summary* that you can use in your custom reports to gain insights about the chatbot processing characteristics. For example, you can get statistics about the number of interactions, the average time, number of messages, efficiency, interaction classification, and social channel utilization.

1. Build and configure a conversational channel:

- a. Create an IVA for Digital Messaging so that users can interact with your application by using the Apple Messages for Business, Facebook, SMS/MMS, Web Messaging, and WhatsApp messaging platforms.

For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).

- b. Define conversational channel behavior.

You define conversational channel behavior by creating stand-alone conversation processes or conversation processes for a case type, adding suggested cases, defining suggested responses, configuring text analyzer settings, and configuring system responses. For more information, see [Defining conversational channel behavior \(on page 172\)](#).

- c. Simulate a conversation and build a chatbot.

You can verify chatbot responses, modify questions and response text, extract data from a conversation, and apply simulation changes. For more information, see [Simulating a conversation and building a chatbot \(on page 199\)](#).

2. Record and train the data for the text analytics model in the IVA channel.

For more information, see [Training the model for the IVA channel \(on page 205\)](#).

3. Troubleshoot the IVA channel.

For more information, see [Troubleshooting the conversational channel \(on page 213\)](#).

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

[Conversational channels \(on page 4\)](#)

Creating a Digital Messaging channel

Expose your essential business services to users through a chatbot on multiple messaging platforms by defining a single Pega Intelligent Virtual Assistant™ (IVA) for Digital Messaging. As a result, you provide users with additional ways to communicate with your application, which improves the overall user experience.



For example, you can create an IVA for a travel agency application. As a result, customers can instantly book their flight by interacting with your chatbot on Apple Messages for Business, Facebook Messenger, and other messaging platforms.



Note: Due to changes by Twitter in the API and the licensing model, direct integration with Twitter is no longer supported in the Digital Messaging channel interface.

You can define custom IVA responses to fit the business needs of your organization and to extract the intent of users with natural language processing (NLP), decisioning, and text analytic capabilities. In the preview console, you can test whether the configured Digital Messaging channel works correctly before moving the channel to a production environment.

If you used a language other than English in Pega Platform™ 8.6 or earlier, for example, Dutch, you can remove all languages from the supported languages list and then add the Dutch language as the first language in the list in the system updated to the 8.7 version, so that it acts as the fallback language. By default, English appears as the first language in the supported languages list when you update to Pega Platform 8.7.

Note:



- The languages that you configure for the Digital Messaging channel also determine the languages in which users can interact in the Web Messaging widget.
- After you update Pega Platform or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.
- Digital Messaging Service is a component hosted on Pega Cloud services that enables an encrypted and secure interaction between a customer's Digital Messaging channel and your application. For more information, see [Digital Messaging Service overview \(on page 118\)](#).

For relevant training materials and to test this feature on a virtual Pega instance, see the [Getting started with Digital Messaging](#) module, the [Digital Messaging channels](#) module, and the [Configuring a chatbot with Digital Messaging](#) challenge on Pega Academy.

1. Create a new channel interface in Pega Platform:

- a. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
- b. In the navigation pane of App Studio, click **Channels**.
- c. In the **Create new channel interface** section, click **Digital Messaging**.
- d. In the **Details** section, in the **Channel interface name** field, enter a name for your channel.
- e. **Optional:** To describe the purpose of the channel, in the **Description** field, enter a short description of your IVA.
- f. In the **Supported languages** list, select the languages in which the chatbot can interact with users.
For each language selected, the application can perform text analysis and provide automatic responses in that language. The first language in the list is the default language, which is the fallback language for the chatbot if no translation is yet available in another language. If no translation is yet available in another language, the system displays placeholder text in the fallback language. To learn more about the localization wizard that you use for translation of chatbot conversations, see [Localizing your application \(on page 170\)](#).
- g. In the channel, click the **Connection** tab.
- h. In the **Template operator ID** list, click the name of an existing Pega Platform operator to use as a template.
The system uses this operator as a template to create all further accounts. Each account has a unique channel identifier set as its ID. The selected operator must at the minimum have the *PegaRULES:User4* access role, taking into consideration the application settings, for example, providing access to the application's cases. For more information, see [Template operator for IVA channel \(on page 171\)](#).
- i. Click **Save**.

2. Configure a Digital Messaging channel for a social messaging platform:

- To enable users to chat in Business Chat, configure the channel for Apple Messages for Business

For more information, see [Setting up the IVA for Apple Messages for Business \(on page 119\)](#).

- To enable users to chat in Facebook Messenger, configure the channel for Facebook.

For more information, see [Setting up the IVA for Facebook Messenger \(on page 121\)](#).



- To enable users to chat with SMS or MMS, configure the channel for SMS/MMS (Twilio).

For more information, see [Setting up the IVA for SMS/MMS \(Twilio\) \(on page 122\)](#).

- To enable users to chat in an embedded window on a webpage and optionally switch to other messaging channels like Facebook Messenger, configure the channel for Web Messaging.

For more information, see [Setting up the IVA for Web Messaging \(on page 131\)](#).

- To enable users to chat in WhatsApp Messenger, configure the channel for WhatsApp.

For more information, see [Setting up the IVA for WhatsApp \(on page 145\)](#).

- To enable users to chat in a native Android app, configure the channel for Android app messaging.

You use Pega Mobile Messaging SDK in your Android app to communicate with the chatbot for your application. For more information, see [Setting up the IVA for Android app messaging \(on page 149\)](#).

- To enable users to chat in a native iOS app, configure the channel for iOS app messaging.

You use Pega Mobile Messaging SDK in your iOS app to communicate with the chatbot for your application. For more information, see [Setting up the IVA for iOS app messaging \(on page 151\)](#).

- To enable users to chat in additional channels, for example, Teams, Webex, or Slack, configure the channel for the Client Channel API. For more information, see [Setting up the IVA for Client Channel API \(on page 153\)](#).

You configure specific settings for a social messaging platform on the **Connection** tab using Digital Messaging Manager that is displayed on a separate page.

[Digital Messaging channel overview \(on page 116\)](#)

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

Digital Messaging channel overview

You create and build a single Digital Messaging channel to ensure that users can interact with your application through multiple social messaging platforms as Pega Intelligent Virtual Assistant™ (IVA). The Digital Messaging channel is quick to create and configure, giving the user a completely new experience. In addition, the Digital Messaging channel makes maintenance of the conversational capability of an IVA much easier.





Note: Due to changes by Twitter in the API and the licensing model, direct integration with Twitter is no longer supported in the Digital Messaging channel interface.

Supported messaging platforms

By defining Pega Platform settings in Digital Messaging Manager, you can create a Digital Messaging channel for the following messaging platforms:



Apple Messages for Business

Send text messages in the Apple Messages for Business app to connect with organizations and ask questions, schedule appointments, and make purchases. You can also send and receive attachments and emojis.



Facebook

Send public posts and private text messages in the Facebook Messenger app. You can also send and receive attachments and emojis.



SMS/MMS (Twilio)

Send short message service (SMS) text messages or Multimedia Messaging Service (MMS) messages using the Twilio communications platform. You can also send and receive attachments and emojis.



Web Messaging

Send text messages in an embedded widget on a webpage for a synchronous chat experience.



WhatsApp

Send text messages in the WhatsApp Messenger app. You can also send and receive attachments and emojis.



Android App Messaging

Send text messages in a native Android app built with Pega Mobile Messaging SDK. You can also send and receive attachments.



iOS App Messaging

Send text messages in a native iOS app built with Pega Mobile Messaging SDK. You can also send and receive attachments.



Client Channel API

Send text messages in additional channels, for example, Teams, Webex, or Slack, by using the Client Channel API.

For relevant training materials, see the [Digital Messaging channels](#) module on Pega Academy.



Note: Digital Messaging Service is a component hosted on Pega Cloud services for Pega Platform that enables encrypted and secure interaction between a customer's Digital Messaging connections listed above and your application. For more information, see [Digital Messaging Service overview \(on page 118\)](#).

[Creating a Digital Messaging channel \(on page 113\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

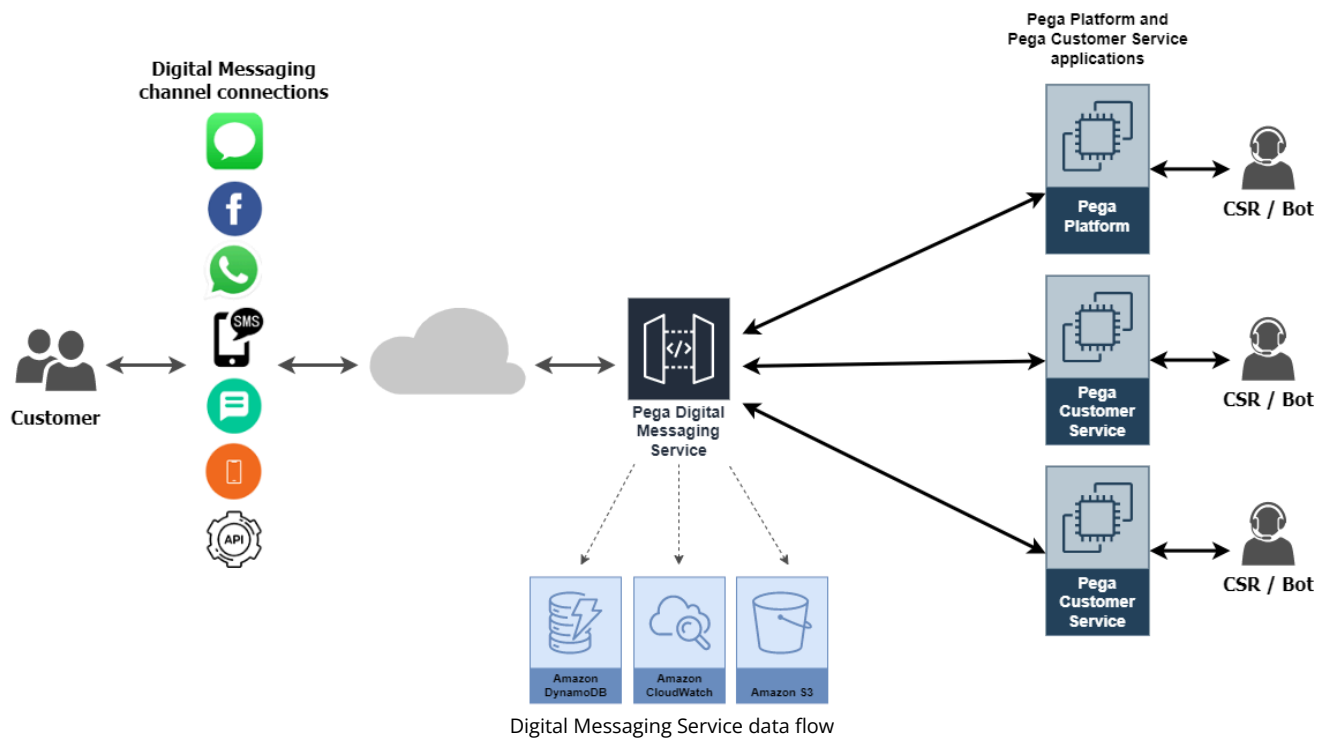
Digital Messaging Service overview

Digital Messaging Service is a component hosted on Pega Cloud services that provides secure and encrypted communication between your Pega Platform application and customers chatting in the Pega Intelligent Virtual Assistant™ (IVA), for example, the Web Messaging, or WhatsApp connection for your Digital Messaging channel.



Note: Due to changes by Twitter in the API and the licensing model, direct integration with Twitter is no longer supported in the Digital Messaging channel interface.

The following figure shows the data flow for Digital Messaging Service between the customer's Digital Messaging channel connections and your Pega Platform or Pega Customer Service application:



The data flow when communicating through Digital Messaging Service consists of the following stages:

1. A customer sends a message to the chatbot or CSR using one of the supported Digital Messaging channel connections, for example, Web Messaging.
2. Digital Messaging Service transforms the incoming message into a Pega format and then sends it to your Pega Platform or Pega Customer Service application.
3. The system sends a reply from the chatbot or CSR back to Digital Messaging Service, which transforms the outgoing message from the Pega format into the correct format for the Digital Messaging channel connection, for example, Web Messaging.
4. The Digital Messaging Service forwards the message to the customer's Digital Messaging channel connection, for example, Web Messaging.

[Configuring on-premises or internal access only environments for Digital Messaging access \(on page 169\)](#)

[Configuring Digital Messaging channel security \(on page 168\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

Setting up the IVA for Apple Messages for Business

Ensure that users can interact with your Pega Intelligent Virtual Assistant™ (IVA) by sending messages from Apple Messages for Business. By conveniently interacting in an everyday messaging platform, users can address their business needs in your application more efficiently. For example, for an online store



application, customers can instantly message your chatbot from their iPhone by using Apple Messages for Business to report problems with their recent product purchase.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts in one place for your application.

For relevant training materials, see the [Configuring Digital Messaging Manager](#) module on Pega Academy.



Note: After you update Pega Platform™ or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.
6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **Apple Messages for Business** icon.
7. On the **New Apple Messages for Business** page, in the **Name** field, enter a label for your Messages for Business account to use in the system.
8. In the **Messages for Business ID** field, enter the identifier for your Messages for Business account. Apple generates a unique identifier when you create an account in the Apple Business Register portal.
9. Click **Save**.

The system displays information that the new Messages for Business account was successfully registered.

10. **Optional:** To display all of your configured accounts for messaging platforms in the **Digital Messaging Manager** window, click **Home**.
11. Close the browser window, and then in the Digital Messaging channel, click **Save**.

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)



Setting up the IVA for Facebook Messenger

Provide users with the ability to interact with your Pega Intelligent Virtual Assistant™ (IVA) in Facebook Messenger. By interacting with your chatbot on Facebook Messenger, users have a convenient, everyday messaging app available to resolve issues or request more help in your application. For example, for a travel agency application, customers can interact with your chatbot in Facebook Messenger to cancel their flight.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts in one place for your application.

For relevant training materials, see the [Configuring Digital Messaging Manager](#) module on Pega Academy.



Note: After you update Pega Platform™ or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.
6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **Facebook** icon.
7. If you are not signed on to your Facebook account in the browser, log in to your account:
 - a. On the Facebook login page, enter the email address and password for your Facebook account.
 - b. Click **Log In**.



Note: The account you use must be an admin of the Facebook page that you want to authorize.

8. In the displayed window, click **Continue as <YOUR NAME>**.
9. In the page selection window for your Facebook page application, select the checkbox for the Facebook page that you will use with your Pega Platform application, and then click **Next**.
10. Review the permissions for your Facebook page, and then click **Done**.



This action ensures that you give permission to your Facebook page app to send and receive messages on behalf of the IVA channel.

The window displays information that your Facebook page is now linked to your Digital Messaging channel.

11. Click **OK**.

The system displays information that the authorization for your Facebook page was successful, and displays a list of all of your configured messaging platforms.

12. **Optional:** To display the Get Started button in the Facebook Messenger, turn on the **Messenger Get Started Button** switch.

The Get Started button displays to the user on the Facebook Messenger welcome screen. When a user clicks the button, the system sends the Get Started message to the conversation and the system grants the chatbot permission to send messages. Because Facebook provides the Get Started button, you cannot customize the button.

The system displays the Get Started button only when users for the first time begin a conversation in Facebook Messenger with your Facebook page. The system displays the Get Started button when both the user starts a conversation directly in Facebook Messenger or the user performs a channel switch from Web Messaging to Facebook Messenger.

Note: The system automatically enables the Get Started button for the connected Facebook pages in Digital Messaging Manager, when you add the page to the channel switch options for your Web Messaging channel.

13. Close the browser window, and then in the Digital Messaging channel, click **Save**.

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Setting up the IVA for SMS/MMS (Twilio)

Ensure that users can interact with your Pega Intelligent Virtual Assistant™ (IVA) by sending SMS or MMS messages from a mobile phone. As a result, you make your business services available to users on a convenient messaging platform as a chatbot, helping to address their important business needs. You need to set up a Twilio account for the phone number that is used by your application as a chatbot. For example,



for a financial bank application, customers can send SMS messages to your chatbot to obtain the current balance on their savings account.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts from one place for your application.



Note: MMS support on Twilio is determined by the mobile phone number. If you create a Twilio account for a mobile phone number that supports MMS, then the system will also support MMS.

For relevant training materials, see the [Configuring Digital Messaging Manager](#) module on Pega Academy.



Note: After you update Pega Platform or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.
6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **SMS** icon.
7. On the **New Twilio SMS** page, in the **Name** field, enter a name for your Twilio account.
8. In the **Number** field, enter the phone number, including the country code, for your Twilio account.
9. In the **Account SID** field, enter the identifier for your Twilio account.
Twilio generates a unique identifier when you create a Twilio account.
10. In the **Auth token** field, enter the authorization token for your Twilio account.
Twilio generates an authorization token when you create a Twilio account.
11. In the **Add country** list, select a country code for which you allow to send messages to your application, and then click **Add**.



Note: Ensure that you align the options you select in this list with the geographic permissions for your Twilio number. In the Twilio portal, you can search for **SMS Geographic Permissions** to configure the locations for the permitted inbound messaging.

The system displays information that the selected country code was successfully updated.

12. **Optional:** To add more country codes, repeat step 11 ([on page 123](#)).
13. Click **Save**.

The system displays information that the new SMS account was successfully registered.

14. In the **Webhook** section, highlight and copy the generated webhook URL for Twilio.
15. **Optional:** To display all of your configured accounts for messaging platforms in the **Digital Messaging Manager** window, click **Home**.
16. Close the browser window, and then in the Digital Messaging channel, click **Save**.
17. Access your account on the Twilio portal, click **Manage Numbers**, and then click **Active Numbers**.
18. In the **Messaging** section, in the field next to the **A Message Comes In** webhook section, paste the webhook URL that you copied in step 14 ([on page 124](#)).
This action ensures that when your Twilio phone number receives inbound SMS or MMS messages, Twilio forwards the messages to the IVA channel.

[Troubleshooting the Digital Messaging channel \(\[on page 217\]\(#\)\)](#)

[Digital Messaging channel overview \(\[on page 116\]\(#\)\)](#)

[Creating a Digital Messaging channel \(\[on page 113\]\(#\)\)](#)

Setting up a Twilio account for a mobile number

Enable verified users to send text messages (SMS or MMS) to a mobile number that you define for a Twilio account. As a result, users can later interact with Pega Intelligent Virtual Assistant™ (IVA) and your application by sending and receiving text messages requesting more information, reporting an issue, or asking for help.

Twilio is a commercial cloud communications platform that lets software developers send and receive text messages, and perform additional communication functions using its web service API. Twilio determines SMS and MMS support for your account based on operator settings for your mobile number. For example, if you create a Twilio account for a mobile number that supports MMS, then Twilio also supports MMS.

1. Create a Twilio account:
 - a. In a web browser, go to the Twilio portal login page, and then click **Sign up**.
 - b. Enter your name, email address, and password, and then click **Start your free trial**.

Twilio sends you an email with a verification link.



- c. Click the verification link in the email, and then enter a mobile number to use with your Twilio account.

Twilio sends an SMS message with a verification code to your phone.

- d. In the Twilio portal, verify the mobile number for your account by entering the verification code.

2. Set up your mobile number for use with the Twilio account:

- a. In the Twilio portal, click **Get a Trial Number** in the **Dashboard** section, and then click **Choose this Number**.

You use the generated account identifier and authentication token by Twilio to later configure an IVA for Digital Messaging so that verified users can send SMS or MMS messages to your Pega Platform application. For more information, see [Setting up the IVA for SMS/MMS \(Twilio\) \(on page 122\)](#).

Twilio displays an account identifier and authorization token for your account, as well as information about whether your mobile number supports SMS and MMS.

- b. In the left panel, click **Phone Numbers > Manage Numbers > Active Numbers**.
- c. In the **Messaging** section, in the **A Message Comes In** field for a webhook, enter the URL generated in the **Webhook** section for your SMS account page in Digital Messaging Manager. For more information, see [Setting up the IVA for SMS/MMS \(Twilio\) \(on page 122\)](#).
- d. Click **Save**.

3. Add test users for your Twilio mobile number:

Before your Twilio mobile number is available in a production environment for a Pega Platform application, only users that you set up in the system as verified callers can send and receive SMS or MMS messages for your account.

- a. In the left panel in the **Phone Numbers** section, click **Verified Caller IDs**.
- b. Click the **Add** icon, and then click **Text me instead**.
- c. Enter the mobile number of a test caller to use with your Twilio account.

Twilio sends a text message with a verification code to this mobile number.

- d. Enter the verification code.

The verified user can now send SMS or MMS messages from this mobile number to your Twilio account.

- e. **Optional:** To add more verified users for your Twilio account, for example, for testing purposes, repeat steps 3.b ([on page 125](#)) through 3.d ([on page 125](#)).

[Creating a Digital Messaging channel \(\[on page 113\]\(#\)\)](#)

Setting up the IVA for Twitter

Ensure that users communicate with your Pega Intelligent Virtual Assistant™ (IVA) by using direct messages on Twitter. As a result, you provide a convenient, everyday way in your application for users to access your business services to obtain assistance or solve an issue.

For example, for a car fee dispute application that a car rental service uses, customers can send messages to your chatbot by using Twitter to question an automatic cancellation fee that is generated by their recent vehicle driver.



Note: Due to changes by Twitter in the API and the licensing model, direct integration with Twitter is no longer supported in the Digital Messaging channel interface.

[Troubleshooting the Digital Messaging channel \(\[on page 217\]\(#\)\)](#)

[Digital Messaging channel overview \(\[on page 116\]\(#\)\)](#)

[Creating a Digital Messaging channel \(\[on page 113\]\(#\)\)](#)

Legacy Webchat to Web Messaging migration best practices

Legacy Webchat was deprecated in Pega Platform™ and Pega Customer Service™ release 8.6 and de-supported in release 8.7. As a replacement, you must use Web Messaging in the Digital Messaging channel interface. Contact your organization's Pega account team to begin the migration process to Web Messaging if not already in progress.



Note: The Pega Legacy Webchat servers will be taken down starting on October 31, 2023.

Important: There are some feature differences between Web Messaging and Legacy Webchat. Some are related to the display of the chat window to your customers and to the functionality available to them. For example, support for forms is being added in Pega Infinity™ '23.



There are also differences resulting from changes in later versions of Pega Platform and Pega Customer Service, in some cases, functionality that was available for Legacy Webchat in 7.x or early 8.x versions is no longer available as of 8.4 or later. For example, URL- based routing is one such difference. For more information on the differences and ways to address them, see [Troubleshooting checklist for the Legacy Webchat to Web Messaging migration \(on page 128\)](#).

To learn about the Legacy Webchat to Web Messaging migration procedure, see [Migrating from Legacy Webchat to Web Messaging](#).

Recommended process to migrate from Legacy Webchat to Web Messaging

Although substantial customization of messaging infrastructure was possible in the past, many rules are now final, and the goal with messaging is configuration over customization. Customization of messaging infrastructure often causes problems when updating and when moving from Legacy Webchat to Digital Messaging

1. Implement Web Messaging and verify end-to-end processes in your lower environments:
 - a. Pega recommends updating to the latest Pega Platform and Pega Customer Service versions prior to enabling Digital Messaging. Follow additional instructions for the version you are updating to, for example, see [Pega Customer Service procedures](#). For more information about extended version support, see [Pega Extended Support program](#).



Important: Legacy Webchat is de-supported in Pega Platform and Pega Customer Service release 8.7. You will not see Legacy Webchat configurations in the system in 8.8 if you do not have an existing implementation. Do not perform a fresh install of 8.8 unless you wish to remove your Legacy Webchat configurations and deploy Digital Messaging from a new application.

- b. Run Pega Customer Service Upgrade Checker to identify and address overrides to final, deprecated and withdrawn rules. If you have overridden rules that have been made final,

this will be a barrier to a successful messaging implementation. For more information, see [Running the Upgrade Checker for Pega Customer Service](#).

- c. Ensure that your messaging implementation is as close to the default settings as possible.
 - d. If you do not have Digital Messaging credentials, contact your organization's Pega account team to have them provisioned for all your applications and environments.
 - e. To preserve your existing bot configurations, copy your Legacy Webchat channel interface to a Digital Messaging channel interface. Alternatively, create a new Digital Messaging channel interface using out-of-the-box settings and configure the chatbot following usual best practices.
 - f. Add a Web Messaging connection to your Digital Messaging channel interface, and then configure your connection settings. For more information, see [Setting up the IVA for Web Messaging \(on page 131\)](#).
 - g. Remove any unnecessary Legacy Webchat code from your web pages where the new Web Messaging widget is installed. This includes removing PegaHelper.js and REDUX scripts.
 - h. Test messaging end-to-end, including escalating and engaging with an agent. If you encounter any issues, follow usual processes to raise them with Pega Support and report them to your organization's Pega account team.
2. After implementing and testing Web Messaging in your lower environments, update your production environment to the same Pega Platform and Pega Customer Service versions, and then implement and test Web Messaging in your production environment, following the best practices listed above.

[Setting up the IVA for Web Messaging \(on page 131\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Troubleshooting checklist for the Legacy Webchat to Web Messaging migration

Use the troubleshooting checklist if you encounter issues and to ensure a seamless and successful migration from Legacy Webchat to Web Messaging. This checklist helps identify and resolve potential challenges, enabling organizations to achieve a smooth transition to their new environment or version of Pega Platform™.



Note: To learn about the Legacy Webchat to Web Messaging migration procedure, see [Migrating from Legacy Webchat to Web Messaging](#).

- For Pega Customer Service release 8.8 and earlier, obtain your Digital Messaging credentials and install your Digital Messaging channel interface in the application built on top of your self-service application implementation layer (*CSSelfService*) to take advantage of out-of-the-box functionality



and rules, such as escalating to a CSR. For Pega Customer Service release '23, create the Digital Messaging channel directly in your implementation layer without the need for the self-service layer. For more information, see [Creating a self-service application](#).

- To identify and resolve overrides to locked rules, and remove deprecated or withdrawn rules in Pega Customer Service, run the Pega Customer Service Upgrade Checker. You can remove an override by copying the overridden Pega Rules to the relevant Ruleset. Pay particular attention to any overridden Data Transform Rules. For more information, see [Running the Upgrade Checker for Pega Customer Service](#).



Note: Some overrides might not be identified by the Upgrade Checker because they are not final. However, modifying them can still cause issues. For example, the *CPMWrapup* flow.

- Check and update the Data Transforms associated with *Declare_CAAppliationSettings*. They might be missing critical elements, for example, the *BotAgentClass*, or certain settings are mapped to the out-of-the-box classes instead of the implementation classes. For information, see [Adopting new settings](#).
- Ensure to install your Digital Messaging Manager ID and Key on the **Application > Definitions > Security** tab of your application. Do not reuse them in another application or instance. Each instance and application must have a unique set of credentials.
- If you cannot connect to the Digital Messaging channel or the **Manage connections** button does not appear to start Digital Messaging Manager, ensure that your channel interface Base URL is secure (uses **https**) and publicly accessible, and that your Template Operator and Access Group are correct. You must have the application where the Digital Messaging channel interface is installed as the default Access Group.
- To test responses, engagement, and escalations to a CSR, connect by using your Web Messaging script installed on a web page, not your channel interface Preview console.
- If you receive an expired notification on agent escalation pop-up in Interaction Portal, ensure that your *D_MCPinteraction* data page is configured to refer to the *Interaction* class group for your implementation application. For more information, see [Configuring implementation class for bot interactions](#).
- Case types used with Legacy Webchat for self-service require configuration to work with Digital Messaging. If the same conversational flow is used in the Digital Messaging channel, update the When Rule to include *.pxActiveChannel == BotAgent*. Alternatively, create a separate parallel process to support the Digital Messaging channel, enabling different behavior in Legacy Webchat and Digital Messaging.
- Some flow elements, for example, flow actions and sections included in paragraph rules, configured for Legacy Webchat are not supported in Digital Messaging.

- Chat server configuration is not required for Digital Messaging Service. Remove chat server settings if present in App Studio or remove any instance in *ChannelServices-Server-Chat*. The system invokes *GetSSOToken* if the chat server is configured.
- Re-save your App Studio chat and messaging settings, for example, timeout and close behavior, messaging attachments.

[Legacy Webchat to Web Messaging migration best practices \(on page 126\)](#)

[Setting up the IVA for Web Messaging \(on page 131\)](#)

Migrating from Legacy Webchat to Web Messaging

Switch to the Digital Messaging interface from your existing Legacy Webchat interface if you want to use the new React-based Web Messaging widget.

1. On the **Security** tab of the application definition in Dev Studio, enter the **Manager ID** and the **Manager Key** in the **Digital Messaging security** section, and then click **Save**.
Make sure you are in the application from your stack in which you are setting up your channels.
2. Navigate to your Legacy Webchat interface.
3. On the right side of the page, click **Actions > Copy channel interface**.
4. In the **Copy channel** pop-up window, click **Bot Agent** to switch from your Legacy Webchat to the Digital Messaging interface.



Note: Creating a copy of the Legacy Webchat channel using the Bot agent option creates a digital messaging interface with built-in conversation flows, case commands, and responses configured in the Legacy Webchat interface.

5. On the **Channel** tab of your new Digital Messaging interface, in the **Set up** section, enter **Template Operator** and **Base URL** (URL for the instance), and then click **Save**.

You can now add your connections to the different Digital Messaging channels through the Digital Messaging Manager.

6. Click **Manage connections** to access the Digital Messaging Manager.
7. In the Digital Messaging Manager, create a Web Messaging connection and configure its appearance.

To display a welcome message to customers when they open the widget for the first time, add the message in the **Display Message** field on the **General** tab of Web Messaging connection. The welcome message configured in the **Behavior** tab of the Digital Messaging interface will not apply.

8. Deploy the Web Messaging widget on the client's website pages by copying the code snippet on the **Installation** tab of your Web Messaging connection and using this snippet to replace the Legacy Webchat Mashup script.

[Creating a Digital Messaging channel \(on page 113\)](#)

[Updating an existing channel to the Digital Messaging channel \(on page 221\)](#)

[Troubleshooting the conversational channel \(on page 213\)](#)

Setting up the IVA for Web Messaging

Enable users to interact with Pega Intelligent Virtual Assistant™ (IVA) from a chat window that you embed in a webpage. As a result, you give users another method to interact with your application and help to address reported issues more efficiently. For example, when customers use an embedded chat window of a website to report a missing part for their purchased mobile phone, your chatbot can respond by asking for more details about the issue and automatically starting a service request case in the application. If you define multiple messaging accounts for your Digital Messaging channel, you can also give users an option to switch from the chat window to another messaging platform, such as Facebook Messenger or WhatsApp, for your application.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts from one place for your application.

If your security team requires JSON Web Token (JWT) authentication for communication between your IVA for Digital Messaging and social messaging platforms, you can configure this alternative security method for your channel. JWT authentication is an industry-standard method that permits the sender to sign a request with a unique token that the receiver later verifies, to ensure that the obtained request is valid and that the token did not expire. For Pega Customer Service™ 8.6 and higher, JWT is the default authentication method. The system automatically selects the JWT authentication method in the Digital Messaging security configuration in your instance, and in Digital Messaging Service.

 **Note:**



- You configure the languages in which the Web Messaging widget can interact with users by defining the list of supported languages in the Digital Messaging channel. For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).
- After you update Pega Platform™ or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.



Note: To learn about accessing website data, for example, tags and cookies for your chatbot, see [Accessing website data when using Pega Intelligent Virtual Assistant \(IVA\) for Web Messaging](#).

For relevant training materials and to test this feature on a virtual Pega instance, see the [Configuring Digital Messaging Manager](#) module and the [Creating a Digital Messaging interface](#) challenge on Pega Academy.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.
6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **Web Messaging** icon.
7. Configure general settings for the Web Messaging widget:
 - a. On the **General** tab, in the **Name** field, enter a name for the Web Messaging account.
 - b. In the **Chat headings** section, in the **Language** list, select a language for the chat heading.
 - c. In the **Chat heading** field, enter a heading that the system displays in the chat window.
 - d. **Optional:** To define a heading in an additional language, click **Add chat heading for another language**, and then repeat steps [7.b \(on page 132\)](#) through [7.c \(on page 132\)](#).
You can define a heading in additional languages based on the list of supported languages for the Digital Messaging channel.
 - e. Click **Save**.

The system displays information that the new Web Messaging account was successfully registered.

8. Configure sound notifications when messages arrive in the chat window:
 - a. In the **Sound notifications** section, in the **Choose when to play** list, select **Don't Play Sound Notifications**, **Play On Every Message**, or **Play Only When Not Visible**.
 - b. If you selected to play a sound for every message or only when the chat window is not visible, in the **Select sound** list, select one of the predefined sounds.
 - c. **Optional:** To hear the audio of the selected predefined sound, click the **Play** icon.

When you enable sound notifications for the chat window, website users can also control their sounds directly from the widget menu by clicking an option to **Mute** or **Unmute**.

9. In the **Widget configuration** section, configure the settings for the widget:
 - a. Select the **Enable emojis picker** checkbox to let users insert a predefined emoji in the chat message during an interaction.
 - b. Select the **Display download transcript option** checkbox to let users download a transcript of the current interaction.
 - c. Select the **Allow customer to upload and send attachments** checkbox to permit users upload and send file attachments during a chat interaction.
 - d. Select the **Display typing indicator** checkbox to display a typing indicator in the chat window to users before the system sends the chatbot response.
 - e. Select the **Display select language option** checkbox to permit users switch to another supported language in the chat window.
When you enable this option, users can switch between each of the supported languages for the Digital Messaging channel. For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).
 - f. Select the **Enable virus scanning for attachments** checkbox to let the system scan file attachments uploaded during a chat interaction for viruses.

10. Adjust the Web Messaging widget appearance.

For more information, see [Adjusting the Web Messaging widget appearance \(on page 135\)](#).

11. **Optional:** To configure a list of additional channels that users can switch to from the chat window:
 - a. Click the **Channel switching** tab.
 - b. In the **Channel switching control placement** section, click **Top right menu** or **Visible in header** to configure where the system displays the option to switch channels.
 - c. Click **Add channel**, and then in the **Add channel** window, in the **Channel** list, select a channel.

d. In the **Connections** section, click the name of your configured channel account, and then click **Submit**.

e. Add more channels by repeating steps 11.c (on page 133) through 11.d (on page 134).

12. **Optional:** To configure the chat context data to display to CSRs when they interact with users:

a. Click the **Context data** tab.

b. In the **General information** section, select the context data to capture during interaction with a user:

Context data	Description
IP address	The IP address of the browser session.
Location	The physical location based on their browser settings.
Browser/device	The web browser and device type.
Current page	The webpage from which the user launched the chat window.

c. Click **Save**.

d. In the **Tags** section, in the **Display Name** field and the **Tag** field, define the tags to add to your website, and then click **Add**.

Use tags to capture custom context data during an interaction with a user so that the system passes this information to a CSR. Custom tags require additional implementation on your website.

e. **Optional:** To add more tags, repeat step 12.d (on page 134).

13. **Optional:** To configure security options for the Web Messaging widget:

a. Click the **Security** tab.

b. In the **Allowed domains** section, in the **New domain** field, enter a domain in which the system can display the Web Messaging widget, and then click **Add**.

 **Note:** If you do not specify any domains, the widget is rendered on any website.

c. **Optional:** To add more domains to the list, repeat step 13.b (on page 134).

- d. In the **Allowed cookies** section, in the **New cookie** field, enter a cookie name that the system allows to pass information between your website and your application.
 - e. **Optional:** To add more cookies to the list, repeat step [13.d \(on page 135\)](#).
 - f. **Optional:** To sign requests while exchanging private data, in the **JWT Secret** section, click **Show JWT Secret**, and then copy the displayed identification code to the clipboard or a file. You can then use the identification code to authorize requests when you exchange private data.
14. **Optional:** To configure proactive chat capability, set up chat triggers for the Web Messaging widget and embed them on your website as JavaScript code snippets.
- With proactive chat your chatbot can engage with users in the Web Messaging chat window by automatically inviting them to a chat conversation based on configured rules and conditions. For more information, see [Configuring proactive chat for Web Messaging \(on page 143\)](#).
15. Click the **Installation** tab, and then copy to clipboard or a text file the displayed script code for your Web Messaging widget.

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Adjusting the Web Messaging widget appearance

Modify the appearance of your Web Messaging widget to align with your organization's preferences and specifications. As a result, you enhance the brand visibility to customers in the Digital Messaging channel, increasing the recognition of your company and its services.

For example, you can showcase a personalized company logo, incorporate custom fonts, and include chat conversation avatars within the Web Messaging widget.

1. In the **Digital Messaging Manager** window, edit your Web Messaging connection, and then click the **Styling** tab.
2. In the **Widget** section, in the **Primary color** palette, pick a color for the Web Messaging widget header.
3. In the **Header text color** palette, pick a color for the Web Messaging widget header text.
4. **Optional:** To display a company logo in the chat window, in the **Company logo** section, select **Display company logo**, click **Upload**, and then select an image to upload.
5. In the **Widget styling configuration** section, configure the settings for the Web Messaging widget:

- a. Select the **Display Powered by Pega** checkbox to display a Powered by Pega footer in the chat window.
- b. Select the **Remove default Widget Launcher** checkbox to hide the chat widget launcher from your webpage.
For example, you can hide the chat widget if you want to replace the default chat bubble with a custom button. You will need to configure a method in your website for the user to start a chat.
- c. Select the **Display avatars in the chat widget** checkbox to display bot and CSR avatars in the chat window.

6. If you selected to display avatars in the widget in the previous step:

- a. In the **Bot avatar type** section, click **Default** or **Custom**.
- b. In the **CSR avatar type** section, click **Default** or **Custom**.
- c. If you selected a custom bot avatar type, click **Upload bot avatar image**, select an image file from your local drive, and then click **Open**.
- d. If you selected a custom CSR avatar type, click **Upload CSR avatar image**, select an image file from your local drive, and then click **Open**.

The system supports the following image file types for bot and CSR avatars: **PNG**, **JPG**, and **JPEG**. The uploaded avatar image must have same width and height, and cannot exceed 500 KB.

7. **Optional:** To use your own font in the chat window:

- a. In the **Custom fonts for widget** section, click **Upload font file**.
- b. Select a font file from your local drive, and then click **Open**.
- c. To upload additional fonts for the chat window, repeat steps [7.a \(on page 136\)](#) through [7.b \(on page 136\)](#).

The system supports the following font file types: **TTF**, **OTF**, **WOFF**, and **WOFF2**. If you upload multiple font files, the browser displays the first supported font from the list for the chat window.

8. **Optional:** Configure the widget position and button type on the webpage:

- a. In the **Widget position** list, select the position on the webpage where you want to display the widget.
- b. In the **Widget button type** list, select the button type for the widget.
- c. If you select a text button type, in the **Text orientation** list, select the text orientation.
- d. If you select a text button type, in the **Button text** field, enter a label for the button, and then click **Save**.

The system updates the widget button type and text orientation on the webpage immediately after you save changes.

9. Click **Save**.

[Setting up the IVA for Web Messaging \(on page 131\)](#)

[Digital Messaging Service overview \(on page 118\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Embedding the Web Messaging widget in a native Pega Platform application

Provide another way for customers to interact with your Pega Platform™ application by embedding the Web Messaging widget in the application's existing harness or a new section. As a result, customers can more efficiently ask for help, find information, or report an issue, by using a Digital Messaging chat window in your application.

When you embed the Web Messaging widget in your application, you use the same JavaScript function that is defined **Installation** tab of Digital Messaging Manager.

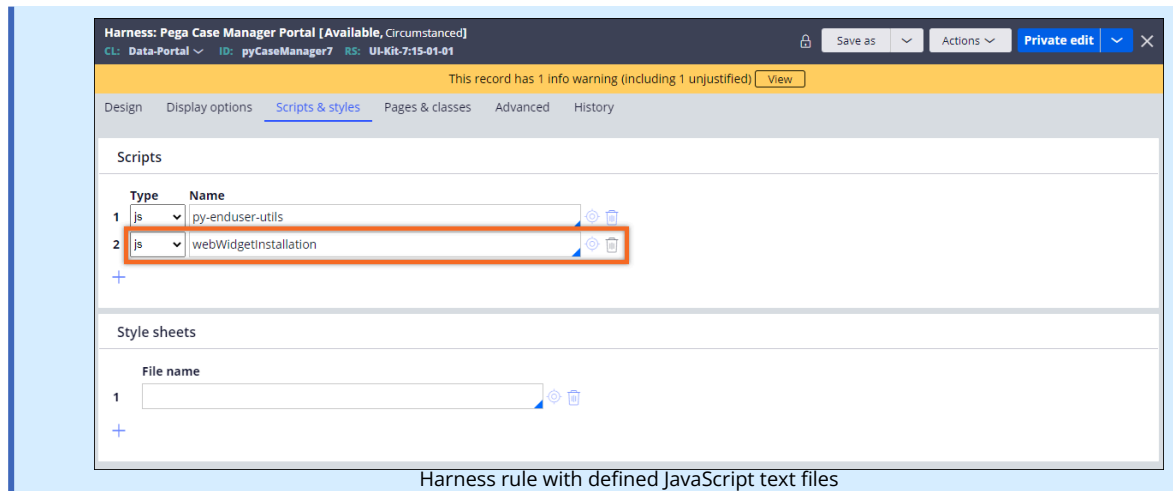
There are two ways to embed the Web Messaging widget in your application.

Include the Web Messaging widget in a harness

- Edit a harness and embed the widget:

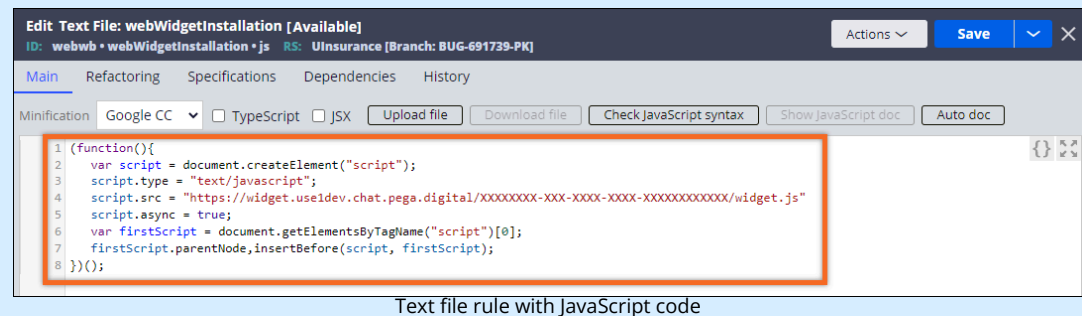
1. Switch to Dev Studio.
2. Open the application that contains the harness where you want to add a Web Messaging widget, and then edit the harness rule.
For more information, see [Harnesses \(on page 113\)](#).
3. In the harness rule, click the **Scripts & styles** tab.
4. in the **Scripts** section, click **Add a row**.
5. In the new row, in the **Name** column, enter `webWidgetInstallation`, and then click **Edit**.

The following figure shows the *pyCaseManager7* harness rule with a new JavaScript text file named **webWidgetInstallation**:



6. Click **Create and open**.
7. In the *webWidgetInstallation* text file rule, on the **Main** tab, in the text area, enter the JavaScript code from the example below to insert the Web Messaging widget.
8. In the JavaScript code, replace the value for `script.src` with the JavaScript function value in the `src` attribute obtained from the **Installation** tab of your Web Messaging channel in Digital Messaging Manager.

The following figure shows the *webWidgetInstallation* text file rule with the JavaScript code needed to embed the Web Messaging widget:



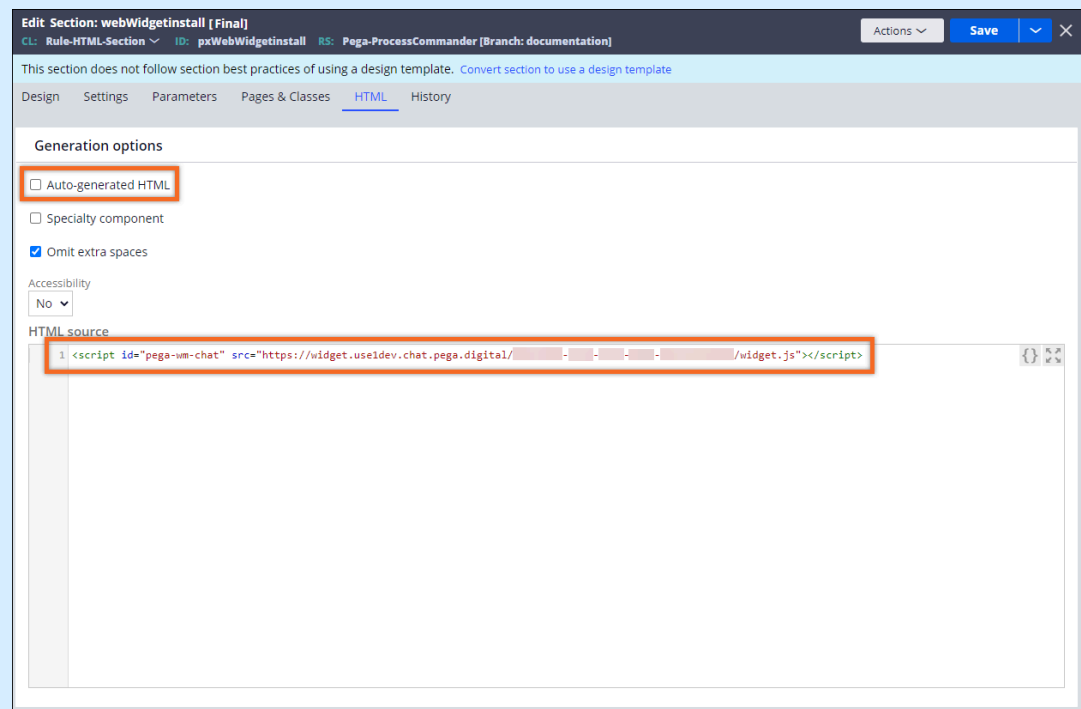
9. Click **Save**.
10. In the harness rule, click **Save as**, and then create a copy of the harness rule in your application ruleset.
11. Switch back to App Studio.

Include the Web Messaging widget in a section

- Create a section and embed the widget:

1. Switch to Dev Studio.
2. In the application where you want to add the Web Messaging widget, create a section rule.
For more information, see [Creating sections \(on page 137\)](#).
3. In the section rule, click **convert to full screen editor**, and then in the **Confirm section conversion** dialog box, click **Submit**.
4. Click the **HTML** tab.
5. Clear the **Auto-generated HTML** checkbox.
6. In the **HTML source** field, enter a script tag from the example below to link to the Web Messaging widget, replacing the `src` attribute value with the JavaScript function obtained from the `src` attribute value from the **Installation** tab in Digital Messaging Manager.

The following figure shows the *webWidgetInstallation* section rule with a script tag defined that calls the Web Messaging widget from Digital Messaging Manager:



Section rule with HTML source code

7. Click **Save**.
8. Switch back to App Studio.

[Setting up the IVA for Web Messaging \(on page 131\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Enabling virus scanning for Web Messaging

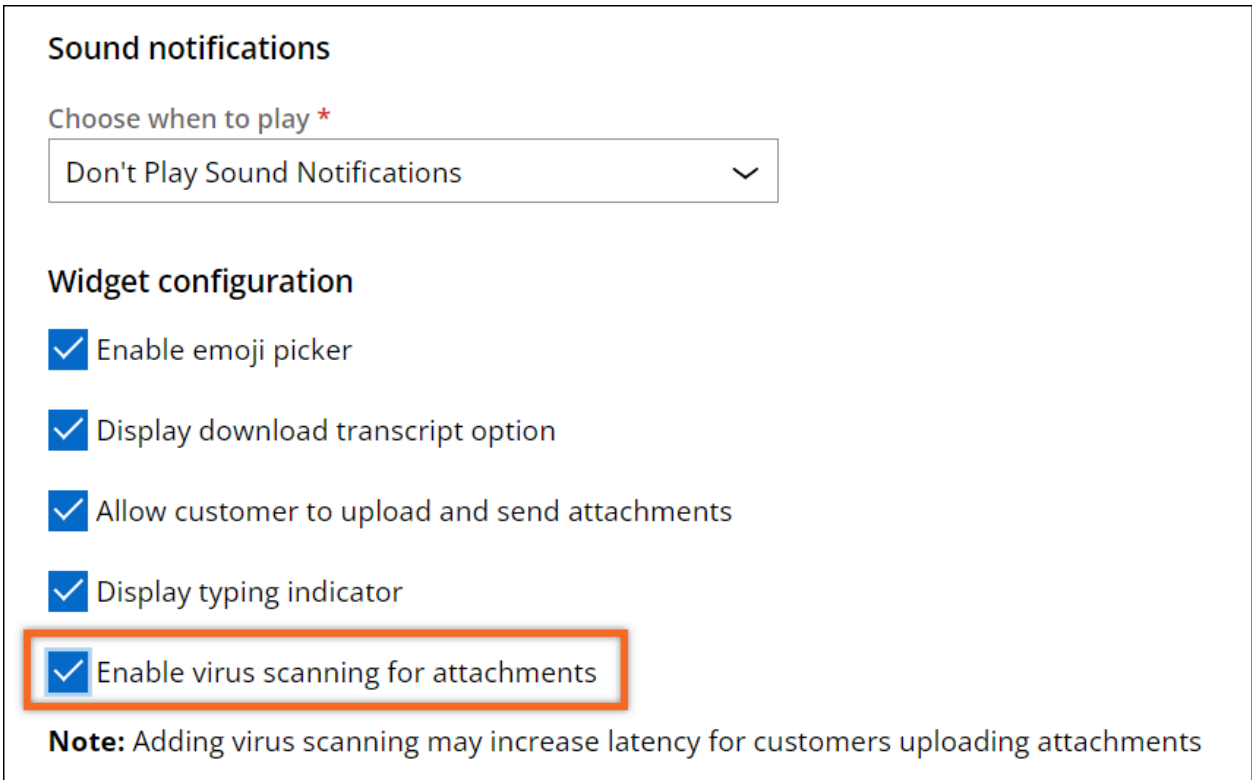
Protect your organization and customers by ensuring that file attachments in inbound and outbound messages do not contain viruses. With this feature, customers and CSRs can confidently interact in the Digital Messaging channel chat window, knowing that the session is more secure and free from potential threats.

The system scans file attachments for viruses added by customers in the Web Messaging chat window and by CSRs in the Pega Customer Service™ desktop application, provided that virus scanning is enabled.

 **Note:** Virus scanning is disabled in the Web Messaging channel connection by default.

1. In your Digital Messaging channel interface, click the **Connection** tab.
2. Click **Manage connections**.
3. In the **Digital Messaging Manager** window, click your Web Messaging connection.
4. On the **General** tab, in the **Widget configuration** section, select the **Enable virus scanning for attachments** checkbox.

The following figure shows where you enable virus scanning of file attachments for the Web Messaging channel connection:



Sound notifications

Choose when to play *

Don't Play Sound Notifications

Widget configuration

- ☒ Enable emoji picker
- ☒ Display download transcript option
- ☒ Allow customer to upload and send attachments
- ☒ Display typing indicator
- ☒ Enable virus scanning for attachments

Note: Adding virus scanning may increase latency for customers uploading attachments

Web Messaging widget configuration in Digital Messaging Manager

5. Click **Save**.

[Native virus scanning in Web Messaging with Digital Messaging \(on page 141\)](#)

[Setting up the IVA for Web Messaging \(on page 131\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Native virus scanning in Web Messaging with Digital Messaging

Digital Messaging Service provides virus scanning capability for attached files during Web Messaging chat interaction for your Digital Messaging channel. Starting with version 8.6, Pega Platform™ lets you instantly scan file attachments, whether sent by customers from the Web Messaging chat window, or by CSRs using the Pega Customer Service desktop application. The virus scanning service uses Cloud Security Storage, a preferred vendor of AWS.



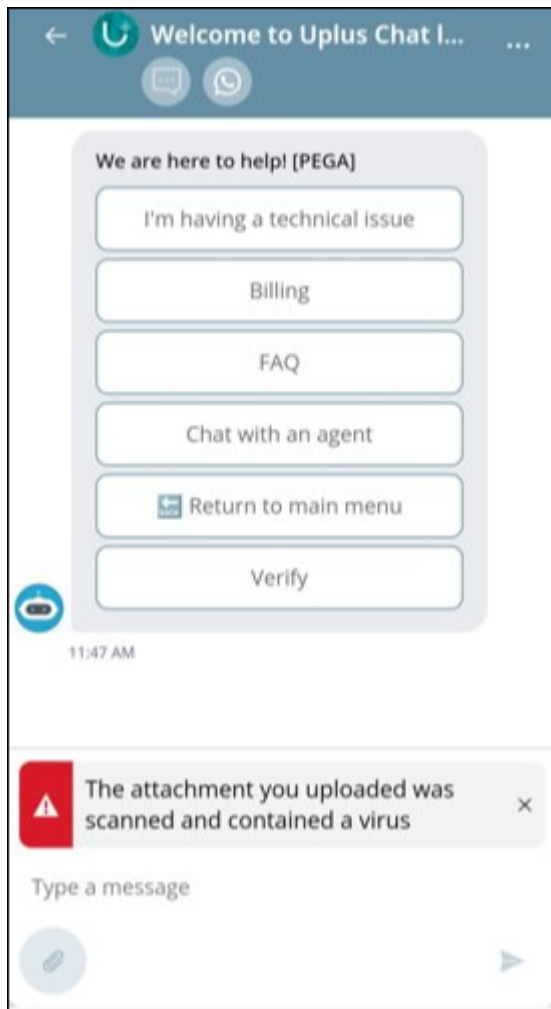
Note: Virus scanning is disabled in the Web Messaging channel connection by default.

Inbound message scanning (customer file attachments)

When you enable the option to scan for viruses, Digital Messaging Service scans file attachments for viruses as customers upload them to the Web Messaging widget. The system starts the virus scanning service as soon as the file attachment enters the S3 bucket (as shown in the diagram), and provides a positive or negative result for the presence of a virus.

- If the service detects a virus in the file attachment, the system automatically removes the virus from the S3 bucket and does not deliver the file attachment to your chatbot or CSR in the chat session. Customers are informed that the system removed their file attachment.
- If the service does not detect a virus, then the system passes the file attachment along to the Pega Platform instance and deletes the attachment from the S3 bucket as usual.

The following figure shows the warning message displayed in the Web Messaging chat window for customers, when a virus is detected after a customer attempts to upload a file:



Warning message in the Web Messaging chat window

Outbound messages scanning (CSR file attachments)

To scan file attachments uploaded by CSRs using Digital Messaging Service, the CSRs must communicate with customers through a Web Messaging widget connection with the virus scanning enabled in the system. When you enable virus scanning in the system, Digital Messaging Service scans outbound file attachments for viruses after they leave the CSR's Pega Customer Service desktop application but before they are sent to the customer in the chat.

If the service detects a virus, the file attachment does not display in the preview console of the CSR's Pega Customer Service desktop application.

[Enabling virus scanning for Web Messaging \(on page 140\)](#)

[Setting up the IVA for Web Messaging \(on page 131\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Configuring proactive chat for Web Messaging

Configure proactive chat in Pega Intelligent Virtual Assistant™ (IVA) for Web Messaging to engage with users in a chat window on your website by automatically inviting them to a conversation based on certain conditions. As a result, you enhance the user experience of your website for users, so that they immediately interact with your chatbot to solve an issue or ask for help, when needed. For example, for an online store, if the total value of a shopping cart reaches a set amount on the website, the system can automatically open the Web Messaging chat window and display a message based on a response command for your IVA. For instance, the chatbot can display information about the available discounts for the dollar amount.

1. In the **Digital Messaging Manager** window, edit your Web Messaging connection, and then click the **Proactive chat** tab.
2. Click **Add trigger**.
3. In the **Add trigger** window, in the **Proactive trigger** field, enter a name for the proactive trigger that you will later reference in the embedded JavaScript code on your website.
4. In the **Display name** field, enter a label to identify the trigger.
5. In the **Bot command** field, enter a chatbot response command that you want to associate with the trigger.
6. Turn on the **Enabled** switch.
7. Click **Add**.
8. Click **Copy code snippet**, and then, in the **Copy code snippet for proactive trigger** window, copy the code snippet to a text file.

The available script from the **Copy code snippet** button is a sample of code that you can embed on the webpage from where the system launches the widget.

9. In the text file, replace the `$trigger_code` variable with the name for the *Proactive trigger* that you specified in step 3 (on page 143).
10. To define and embed more triggers on your website, repeat steps 2 (on page 143) through 9 (on page 144).

[Setting up the IVA for Web Messaging \(on page 131\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Passing website language using PegaChatTags

Improve the global customer experience and maintain a consistent and professional appearance of your Web Messaging channels by implementing a PegaChatTags script. This script passes your website's preferred language to determine the language of the Web Messaging widget when a customer accesses it.

The Web Messaging widget follows a specific sequence and conditional logic to determine the language, continuing until a value is found. In the list below, if the conditions in the first three options are not met, the widget reverts to the **Primary Business Language** configured in your Digital Messaging Manager Web Messaging settings.

1. Menu selection

If you enable the **Allow users to select a language from menu** checkbox on the **Language** tab of your connection in Digital Messaging Manager, customers can choose their preferred language from a drop-down menu.

2. lang parameter in the PegaChatTags script

You can automatically pass a language in the website by using the **lang** parameter in the *PegaChatTags* script.

3. Browser language

You set this option at the browser level and use this option if the Digital Messaging Service supports it. Customers can set this language in their specific browser, for example, in Google Chrome, by going to **Settings > Languages**.

4. Primary Business Language

You set this value on the **Language** tab of your connection in Digital Messaging Manager. The system selects this as a fallback option, if none of the first three conditions are met.

Install either of the following scripts in your webpage from which the Web Messaging widget is launched:

```
<script language="javascript">
  var PegaChatTags = {
```



```
"lang": "{ Placeholder for Customer Info }"
}
</script>
```

For example, in the above script, you can use the **lang** parameter to select the Spanish language as the preferred language for this option, as shown in the definition below:

```
"lang": "es"
```

Alternatively, you can also use the following script that uses the Document Object Model (DOM) to pass the browser language:

```
<script language="javascript">
    var PegaChatTags = {lang: document.documentElement.lang}
</script>
```

[Setting up the IVA for Web Messaging \(on page 131\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Setting up the IVA for WhatsApp

Provide users with an additional messaging platform to access your Pega Intelligent Virtual Assistant™ (IVA) that is convenient and widely available. By interacting with your chatbot in WhatsApp Messenger, users can efficiently resolve their issues, request help, or report problems within your business application. For example, when a customer requests a car insurance quote by interacting in WhatsApp Messenger with your chatbot, the system can respond by asking for more information, and then automatically start an Insurance Quote business case in your application.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts in one place for your application.

For relevant training materials, see the [Configuring Digital Messaging Manager](#) module on Pega Academy.



Note: After you update Pega Platform™ or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.



4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.
6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **WhatsApp** icon.
7. In the **New Twilio WhatsApp** page, in the **Name** field, enter a name for your WhatsApp account.
8. In the **Number** field, enter a WhatsApp number, including the country code.
9. In the **Account SID** field, enter the identifier for your Twilio account.
Twilio generates a unique identifier when you create a Twilio account.
10. In the **Auth token** field, enter the authorization token for your Twilio account.
Twilio generates an authorization token when you create a Twilio account.
11. In the **Webhook** section, select and copy the generated webhook URL for WhatsApp.
12. Click **Save**

The system displays information that the new Twilio WhatsApp account was successfully registered.

13. **Optional:** To display all of your configured accounts for messaging platforms in the **Digital Messaging Manager** window, click **Home**.
14. Close the browser window, and then in the Digital Messaging channel, click **Save**.
15. Define the webhook URL for the WhatsApp number in your Twilio account by performing the following steps:
 - a. Log in to your Twilio account.
 - b. Click the **More** icon, and then click **Programmable Messaging > Senders > WhatsApp Senders**.
 - c. In the displayed list, select a WhatsApp number, and then click **Edit**.
 - d. In the **Endpoint Configuration** section, in the **Webhook URL for incoming messages** field, paste the webhook URL that you copied in step 11 ([on page 146](#)).
This action ensures that when your phone number receives inbound messages from WhatsApp Messenger, Twilio forwards the messages to the IVA channel.
 - e. To add additional WhatsApp numbers for your Twilio account to the webhook URL, repeat steps 15.c ([on page 146](#)) through 15.d ([on page 146](#)).
 - f. Click **Close**.

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)



Adding a WhatsApp number to your Twilio account

Enabling users to interact with Pega Intelligent Virtual Assistant™ (IVA) using WhatsApp Messenger requires first configuring a WhatsApp number in the Twilio portal. The WhatsApp number that you connect to your Twilio account must be upgraded to a paid account, to ensure that you can configure the IVA for WhatsApp in Pega Platform so that users can report issues or ask for help in your application.

Creating a WhatsApp account for business purposes is a multi-step process that uses information from your organization, Facebook (the owner of WhatsApp), Twilio, and Pega.

1. Set up a sandbox for your WhatsApp account and accept the terms of service on the Twilio portal. For more information, see *Help Center section for Whats App* on the Twilio console.
2. Set up your WhatsApp business profile and add an existing mobile number to the profile. You can also read the Twilio guide describing this procedure on the Twilio portal. For more information, see the *Setting up a business profile* section in the *Twilio API for WhatsApp Best Practices and FAQs* guide on the Twilio website.
3. Send a request to Twilio to enable your official WhatsApp business profile to access Twilio numbers. When you apply for an official account to access Twilio numbers for WhatsApp, the application form includes the **I wish to provision this account with a third party vendor** field. Use this field when delegating the management of your WhatsApp business profile in Twilio to a third-party organization. In most situations you will not be delegating account management, so you can leave this field empty. For more information, see *Request access to enable your Twilio numbers for WhatsApp* on the Twilio website.

After approximately two weeks, Twilio approves your WhatsApp business profile.

4. Find and connect the WhatsApp number to your WhatsApp business profile. For more information, see *Connect your Twilio number to your WhatsApp business profile* on the Twilio website.
5. **Optional:** To be able to send unsolicited WhatsApp messages outside of the last 24 hours to users, create 24-hour window templates. For more information, see [Optional WhatsApp configuration for your Twilio account \(on page 148\)](#).
6. **Optional:** To obtain the official business account status for your WhatsApp business profile, Twilio must verify your WhatsApp account. For more information, see [Optional WhatsApp configuration for your Twilio account \(on page 148\)](#).

[Best practices when using an IVA for WhatsApp \(on page 148\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)



Optional WhatsApp configuration for your Twilio account

After you successfully add a WhatsApp number to your Twilio account, you can perform additional configuration to enhance how Pega Intelligent Virtual Assistant™ (IVA) interacts with users on WhatsApp Messenger. Performing these optional steps improves the functionality of the WhatsApp profile used by the IVA, for example, by letting the system send messages outside of the saved conversation in WhatsApp Messenger.

Create 24-hour window templates

To prevent spam in the system, the WhatsApp social messaging platform generally prevents you from sending unsolicited messages to users. In other words, messages that are outside of a conversation thread. The WhatsApp social messaging platform states that you cannot send a message to a user unless that user has sent a message to you within the last 24 hours. In order to send messages to users outside of the last 24 hours, you can create an exception to this rule for your WhatsApp profile by sending notifications based on templates. For more information, see *Send WhatsApp notification messages with templates* on the Twilio website.

Obtain Official Business Account status for WhatsApp

You can optionally obtain an Official Business Account status for your WhatsApp profile. Your Twilio profile then features a WhatsApp badge of approval, which works similarly to the verified account designation on Twitter. The Official Business Account status is not a requirement, as you do not need an official business account for an IVA to interact with users on WhatsApp Messenger. To learn how to obtain the Official Business Account status for your WhatsApp profile, see the *WhatsApp Verified Accounts* section in the *Twilio API for WhatsApp* guide on the Twilio website.

[Best practices when using an IVA for WhatsApp \(on page 148\)](#)

[Setting up the IVA for WhatsApp \(on page 145\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Best practices when using an IVA for WhatsApp

After you successfully configure Pega Intelligent Virtual Assistant™ (IVA) for Digital Messaging to interact with users through WhatsApp Messenger, and define the chatbot behavior, there are several best practices to consider before you train and test the text analytics model, and move the system to a production environment. By following these guidelines, you can troubleshoot some issues related to the WhatsApp service, for example, when the IVA does not receive a response from users or cannot send large image files in WhatsApp Messenger.

Keep in mind the following important considerations when working with the IVA for WhatsApp:



- WhatsApp does not guarantee the availability of their service and might, from time to time, experience outages and delays that impact the delivery of messages to or from your customers. For example, on rare occasions, messages can arrive late to your Pega Platform queues because the WhatsApp service is sending them late. There might also be rare cases when you are unable to send any messages by using WhatsApp Messenger, due to problems with the WhatsApp service. In these situations, contact Pega Support for updates on changes in the WhatsApp service status. For more information, go to [Pega Support](#).
- By default, you cannot send a message to users in WhatsApp Messenger unless they have previously sent a message to you within the last 24 hours. You can create an exception to this rule for your WhatsApp profile by sending notifications based on templates. For more information, see the *Create 24-hour window templates* section in [Optional WhatsApp configuration for your Twilio account \(on page 148\)](#).
- You can send and receive image files in WhatsApp Messenger if the image file is in one of the following formats: **GIF**, **JPEG/JPG**, **MP4**, and **PNG**.
- The maximum size of image files that you can send or receive in WhatsApp Messenger is 5 MB. The system automatically discards images that are larger than 5 MB.
- A user can, at any time, delete a chat conversation with you in WhatsApp Messenger. If you send messages to a user that has deleted a conversation with you, you will not receive a response.

[Setting up the IVA for WhatsApp \(on page 145\)](#)

Setting up the IVA for Android app messaging

Enable users to interact with Pega Intelligent Virtual Assistant™ (IVA) from a native Android app by using Pega Mobile Messaging SDK. As a result, you make your mobile app user experience seamless and more interactive. For example, users can use your Android app to quickly interact with a chatbot that provides support for a financial organization, helping them to resolve issues with loan applications.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts from one place for your application.

If your security team requires JSON Web Token (JWT) authentication for communication between your IVA for Digital Messaging and social messaging platforms, you can configure this alternative security method for your channel. JWT authentication is an industry-standard method that permits the sender to sign a request with a unique token that the receiver later verifies to ensure that the obtained request is valid and that the token did not expire.



Note: After you update Pega Platform™ or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.



1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.
6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **Android App Messaging** icon.
7. On the **General** tab, in the **Name** field, enter a name for the Android App Messaging connection.
8. Click **Save**.
9. **Optional:** To configure the chat context data to display to CSRs when they interact with users:
 - a. Click the **Context data** tab.
 - b. In the **General information** section, select the context data to capture during interaction with a user:

Context data	Description
IP address	The IP address of the mobile device.
Location	The physical location of the mobile device based on GPS settings.
Device/OS	The mobile operating system and device type.

- c. Click **Save**.
10. **Optional:** To configure security options for Android app messaging:
 - a. Click the **Security** tab.
 - b. **Optional:** To sign requests during communication between IVA for Digital Messaging and your Android app, in the **JWT Secret** section, click **Show JWT Secret**, and then use this key for your token during JWT authentication.
You select the JWT authentication method for the Digital Messaging channel in the application rule. For more information, see [Configuring Digital Messaging channel security \(on page 168\)](#).
11. Click the **Installation** tab.
12. Copy the generated values for **Api id** and **Api url** to a file.
13. Click **Download Android SDK** to download Pega Mobile Messaging SDK for Android to your computer.
Pega Mobile Messaging SDK helps you embed web messaging in your native Android app.
14. **Optional:** To configure the ability to receive messages in your Android app when a user is away:



Note: Pega Mobile Messaging SDK for Android does not send push notifications directly to devices.

- a. Click the **Push notifications** tab.
- b. In the **URL** field, enter the Webhook URL for your Android app.
- c. In the **Custom headers** section, in the **Header key** field, enter a header key.
- d. In the **Header value** field, enter a header value.
- e. Click **Add**.
- f. To add more headers, repeat steps [14.c \(on page 151\)](#) through [14.e \(on page 151\)](#).

[Setting up the IVA for iOS app messaging \(on page 151\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Setting up the IVA for iOS app messaging

Ensure that users can message with Pega Intelligent Virtual Assistant™ (IVA) from a native iOS app by using Pega Mobile Messaging SDK. As a result, you make your mobile app user experience seamless and more interactive. Users can instantly interact with the chatbot from your iOS app to get support with a car rental service, for example, to request more information about the availability of cars in a city.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts from one place for your application.

If your security team requires JSON Web Token (JWT) authentication for communication between your IVA for Digital Messaging and social messaging platforms, you can configure this alternative security method for your channel. JWT authentication is an industry-standard method that permits the sender to sign a request with a unique token that the receiver later verifies, to ensure that the obtained request is valid and that the token did not expire.



Note: After you update Pega Platform™ or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.



3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.
6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **iOS App Messaging** icon.
7. On the **General** tab, in the **Name** field, enter a name for the iOS App Messaging connection.
8. Click **Save**.
9. **Optional:** To configure the chat context data to display to CSRs when they interact with users:
 - a. Click the **Context data** tab.
 - b. In the **General information** section, select the context data to capture during interaction with a user:

Context data	Description
IP address	The IP address of the mobile device.
Location	The physical location of the mobile device based on GPS settings.
Device/OS	The mobile operating system and the device type.

- c. Click **Save**.
10. **Optional:** To configure security options for iOS app messaging:
 - a. Click the **Security** tab.
 - b. **Optional:** To sign requests during communication between IVA for Digital Messaging and your iOS app, in the **JWT Secret** section, click **Show JWT Secret**, and then use this key for your token during JWT authentication.
 You select the JWT authentication method for the Digital Messaging channel in the application rule. For more information, see [Configuring Digital Messaging channel security \(on page 168\)](#).
11. Click the **Installation** tab.
12. Copy the generated values for **Api id** and **Api url** to a file.
13. Click **Download iOS SDK** to download Pega Mobile Messaging SDK for iOS to your computer.
 Pega Mobile Messaging SDK helps you embed web messaging in your native iOS app.
14. **Optional:** To configure the ability to receive messages in your iOS app when a user is away:



Note: Pega Mobile Messaging SDK for iOS does not send push notifications directly to devices.

- a. Click the **Push notifications** tab.
- b. In the **URL** field, enter the Webhook URL for your iOS app.
- c. In the **Custom headers** section, in the **Header key** field, enter a header key.
- d. In the **Header value** field, enter a header value.
- e. Click **Add**.
- f. To add more headers, repeat steps [14.c \(on page 153\)](#) through [14.e \(on page 153\)](#).

[Setting up the IVA for Android app messaging \(on page 149\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Setting up the IVA for Client Channel API

Use the Client Channel API to securely communicate between your Pega Platform™ application, your integration layer, and third-party client channels, for example, Teams, Webex, or Slack. As a result, you make Pega Intelligent Virtual Assistant™ (IVA) for Digital Messaging more accessible by engaging with your customers in their preferred social messaging channels. The Client Channel API provides you with a Digital Messaging Service endpoint to which you can send payload requests in JSON format. The payload requests indicate an action during a chat conversation, for example, when a customer in the client channel sends a message. Similarly, you can set up an endpoint for your integration layer to receive payload requests from the Digital Messaging Service, for example, when the chatbot or CSR sends a reply or is busy typing.

You access Digital Messaging Manager in the Digital Messaging channel to configure individual messaging accounts in one place for your application. For relevant training materials, see the [Configuring Digital Messaging Manager](#) module on Pega Academy.



Note: After you update Pega Platform or Pega Customer Service™, to use the latest Digital Messaging Manager features, save your existing Digital Messaging channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the Digital Messaging channel, click the **Connection** tab.
5. Click **Manage connections**.



6. In the **Digital Messaging Manager** window, click **Add Connection**, and then click the **Client Channel API** icon.
7. In the **New Client Channel API** section, complete the following fields:
 - a. In the **Name** field, enter a unique name for your channel.
 - b. In the **Client webhook** field, enter the URL that you previously configured in your integration layer for your client channel. For more information, see [Client Channel API \(on page 155\)](#).

Note: You must configure the webhook URL so that the URL is publicly accessible (off VPN) to enable communication with Digital Messaging Service. If your integration layer is behind a firewall or otherwise limits access to internal-only users and components, you must set up the webhook URL so that the URL is reachable publicly. Ensure that the URL string starts with "https".

8. Click **Save**.

Digital Messaging Service creates a new channel connection and displays the following fields:

Connection ID (JWT issuer)

A unique ID assigned to your connection. Use this ID to generate a JSON Web Token (JWT) that is required for transmitting messages.

Client Webhook

The URL that you specify when creating the connection. This URL is the publicly accessible target endpoint for your client channel to receive messages from Digital Messaging Service and your chatbot and CSR.

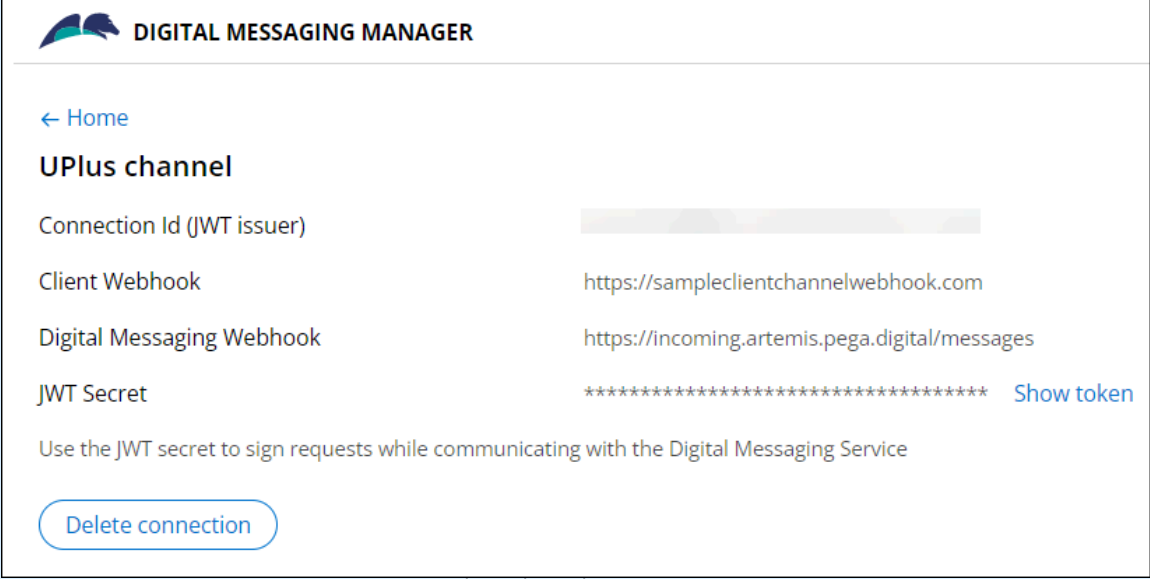
Digital Messaging Webhook

The endpoint URL on Digital Messaging Service. The service uses this URL to receive messages from your client channel sent through your integration layer.

JWT Secret

The service uses the JWT Secret value to authenticate the communication between your integration layer and Digital Messaging Service. Use the JWT Secret value to generate the JWT that is required for transmitting messages.

The following figure shows the sample values generated for a configured Client Channel API connection:



DIGITAL MESSAGING MANAGER

[← Home](#)

UPlus channel

Connection Id (JWT issuer)	[Redacted]
Client Webhook	https://sampleclientchannelwebhook.com
Digital Messaging Webhook	https://incoming.artemis.pega.digital/messages
JWT Secret	***** Show token

Use the JWT secret to sign requests while communicating with the Digital Messaging Service

[Delete connection](#)

Client Channel API settings

Your customers can send and receive messages from the chatbot or an agent in the same way that they send and receive messages using other connection types for the Digital Messaging channel.

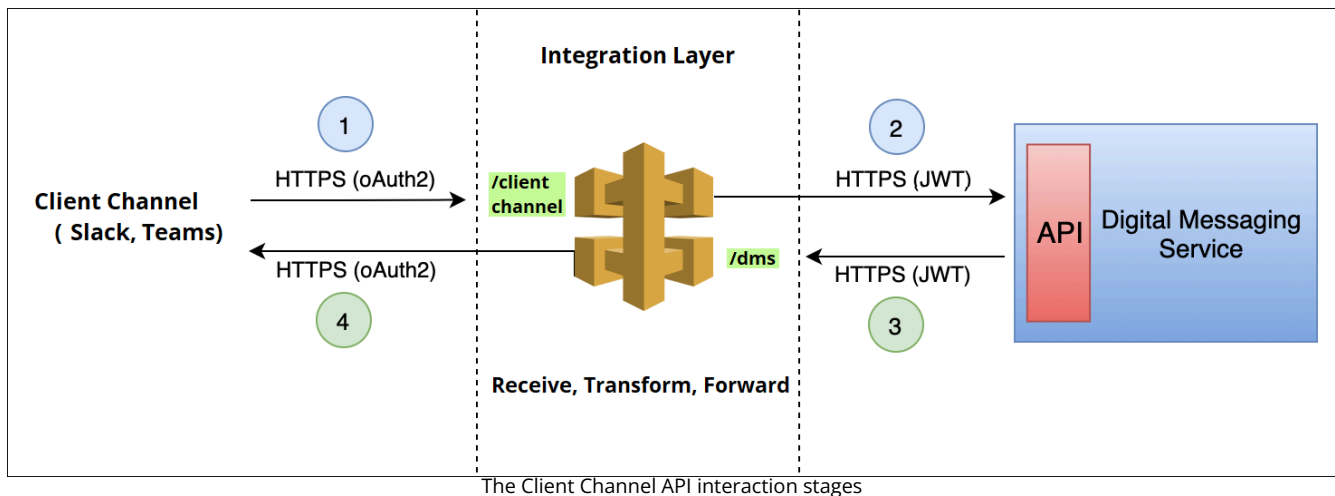
[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Client Channel API

The Client Channel API provides you with a standardized method to exchange messages between several types of third-party client channels, for example, Teams, Webex, or Slack, and Digital Messaging Service.

The process of sending and receiving messages as JSON payload requests using the Client Channel API consists of four stages:



1. Your client channel sends messages to the integration layer that you configure in premises (back-end), using the communication method that is specific to the type of client channel.
2. In your integration layer, transform the received client channel payload into the format that is required by the Client Channel API. Then, pass the message to the endpoint for the Digital Messaging Service.

This endpoint is the Digital Messaging Webhook URL that the system specifies in your Digital Messaging Manager and Client Channel API settings, for example: `https://incoming.artemis.pega.digital/messages`

The Digital Messaging Service then forwards the message to your Pega Platform application.

3. Your Pega Platform™ application returns a response to the Digital Messaging Service which then forwards the response to the endpoint that you specify for your integration layer in your Digital Messaging Manager Client Channel API settings, for example: `https://sampleclientchannelwebhook.com`. You must configure the webhook URL so that the URL is publicly accessible (off VPN) to enable communication with Digital Messaging Service. If your integration layer is behind a firewall or otherwise limits access to internal-only users and components, you must set up the webhook URL so that the URL is reachable publicly.
4. Your integration layer transforms the received Digital Messaging Service payload into the required format by your client channel, and then passes the message to the client channel endpoint.



Note: To learn about the supported format of the JSON payloads for the Client Channel API, see [Client Channel API payload requirements \(on page 158\)](#).

Supported message types

The system supports the following features during communication from a third-party client channel to Digital Messaging Service:

- Text messages
- Typing indicators
- Attachments
- Postbacks (HTTP POST to the same place)
- End session event (customer ended the conversation)
- Context data

The system supports the following features while communicating from Digital Messaging Service to a third-party client channel:

- Text messages
- Typing indicators
- Attachments
- List pickers
- Links or buttons
- Carousels
- End session event (CSR or chatbot ended the conversation)

[Creating a Digital Messaging channel \(on page 113\)](#)

[Setting up the IVA for Client Channel API \(on page 153\)](#)

[Client Channel API guidelines \(on page 157\)](#)

Client Channel API guidelines

Use the guidelines in this section to help you design the best possible third-party client channel integration with the Digital Messaging channel.

Consider the following best practices for your client channel integration with the Client Channel API:

- The Client Channel API does not provide a direct integration to any client channels, for example, Slack or Teams.
- You cannot create an integration in your local environment that directly connects to your client channel and Digital Messaging Service. You must build a custom integration layer that meets the endpoint and payload requirements for both your client channel and the Client Channel API, in order to pass messages between them.

- Each client channel has unique setup and configuration guidelines for integrating with an API. Consult the documentation for the third-party client channel for specific instructions on how to configure the integration.
- You must configure the URL endpoints on your local environment to enable the message delivery between your client channel and the Client Channel API. The endpoint for the Client Channel API is displayed in your Digital Messaging Manager settings. For your client channel endpoints, consult API instructions that are specific to your third-party client channel, in order to create your implementation endpoints. You must configure the webhook URL so that the URL is publicly accessible (off VPN) to enable communication with Digital Messaging Service. If your integration layer is behind a firewall or otherwise limits access to internal-only users and components, you must set up the webhook URL so that the URL is reachable publicly.
- You must secure all endpoints using the Transport Layer Security (TLS) protocol, by using `https`.
- The system uses JSON Web Tokens (JWT) to authenticate the communication in both directions, between your local environment and Digital Messaging Service. You must generate a valid JWT to include in each request to send.
- The Client Channel API does not track chat sessions. You must handle session control in your integration layer or the client channel.

[Creating a Digital Messaging channel \(on page 113\)](#)

[Setting up the IVA for Client Channel API \(on page 153\)](#)

[Client Channel API \(on page 155\)](#)

Client Channel API payload requirements

To communicate between your Digital Messaging channel and a third-party client channel for example, WebEx or Teams, you use the Client Channel API by sending and receiving JSON data in a specific payload format. As a result, your integration layer can interact with Digital Messaging Service and your third-party client channel in the correct format, providing a seamless experience to the customer. You can configure different payload types in the same API in your integration layer.

Authorization

To send and receive JSON requests between the integration layer and Digital Messaging Service, JSON payload headers must include the following information:

Authorization token

The "Bearer {JWT}" value.

The `connection_id` value

The unique ID of the connection for your client channel.



Your integration layer must generate an authorization token that you include in each JSON request. Use the *connection_id* value as the JSON Web Token (JWT) issuer and sign the JWT using the value for JWT Secret, as shown in the following sample code:

```
var jwt = require("jsonwebtoken")
const token = jwt.sign({iss: "connection_id"}, 'jwt_secret');
console.log(token);
```

After you successfully set up your Pega Intelligent Virtual Assistant™ (IVA) and add a Client Channel API connection in the Digital Messaging Manager, the system stores the *connection_id* and JWT Secret values in your Client Channel API settings. For more information, see [Setting up the IVA for Client Channel API \(on page 153\)](#).



Note: For security reasons, each authorization JWT token is valid only for 5 minutes from the time your integration layer issues the token. This expiry period is fixed.

Incoming payload format

To send request payloads from your integration layer to Digital Messaging Service, invoke

POST {DIGITAL_MESSAGING_WEBHOOK} with a JSON request body, for example: **POST https://incoming.artemis.pegasystems.com/messages**

When a customer performs any of the following actions in the client channel, send a request payload from your integration layer:

- Sends a message
- Is busy typing their message
- Ends the chat conversation

Customer message

The following is the JSON body format of the request payload for the customer message that your integration layer sends to Digital Messaging Service:

```
Body (JSON)
{
  "type": "text",
  "customer_id": "string",
  "customer_name": "string",
  "message_id": "string",
  "text": ["string"],
```



```

"postback": "string",
"attachments": [
  {
    "url": "string"
  }
],
"context_data": {
  "string": "string"
}
}

```

The following table describes the parameters passed in the Customer message payload request:

Parameter	Description	Type
<i>type</i>	The static value "text".	Required
<i>customer_id</i>	The ID of the customer sending the message. The ID is a unique value that is provided by the client channel.	Required
<i>customer_name</i>	The name of the customer sending the message.	Optional
<i>message_id</i>	The ID of the message. The ID is a unique value that is provided by the client channel and must be different for every message.	Required
<i>text</i>	An array of strings that specify messages from the customer. See also the note below.	Optional
<i>postback</i>	The payload value of the menu or carousel item that is selected. See also the note below.	Optional
<i>attachments[].url</i>	Temporary private URL that references a file attachment. See also the note below.	Optional
<i>context_data.key</i>	The context data key.	Optional
<i>context_data.value</i>	The context data value.	Optional



Note: If the JSON payload that you send contains empty fields for the *text*, *attachment* and *postback* parameters, then Digital Messaging Service will not receive the message.

The system uses the value passed in the *postback* parameter as a response to a menu or a carousel selection, and the value must match the menu or carousel item payload value that is selected by the customer. The system uses the key and value pairs to pass context data within the message.



The attachment URL must be a valid downloadable value, for example, <https://samplesite.com/filename.pdf>. The system receives *message_id* as the *source_id* and uses the value as the *pySourceID* in your application.

Typing indicator

The following code is the JSON body format of the request payload that your integration layer sends to Digital Messaging Service when the customer is typing a message:

```
Body (JSON)
{
  "type": "typing_indicator",
  "customer_id": "string",
}
```

The following table describes the parameters passed in the Typing indicator payload request:

Parameter	Description	Type
<i>type</i>	The static value "typing_indicator".	Required
<i>customer_id</i>	The ID of the customer that is sending the message. The ID is a unique value that is provided by the client channel.	Required

Customer ended the conversation

The following is the JSON body format of the request payload that your integration layer sends to Digital Messaging Service when the customer ends the conversation:

```
Body (JSON)
{
  "type": "customer_end_session",
  "customer_id": "string",
}
```

The following table describes the parameters passed in the Customer ended the conversation payload request:

Parameter	Description	Type
<i>type</i>	The static value "customer_end_session".	Required
<i>customer_id</i>	The ID of the customer that is sending the message. The ID is a unique value that is provided by the client channel.	Required

Outgoing payload format

To receive request payloads from Digital Messaging Service in your integration layer, the system uses `POST {CLIENT_WEBHOOK}` to send a JSON request body, for example: **POST** <https://sampleclientchannelwebhook.com>

You receive a request payload in your integration layer when the chatbot or CSR performs one of the following actions in your Digital Messaging channel:

- Sends a message
- Displays a menu
- Displays a carousel
- Sends a link
- Is busy typing in the chat window
- CSR ends the chat conversation

CSR message

The following is the JSON body format of the request payload that your integration layer receives from Digital Messaging Service when the chatbot or CSR sends a message:

```
Body (JSON)
{
  "type": "text",
  "customer_id": "string",
  "message_id": "string",
  "csr_name": "string",
  "text": "string",
  "attachments": [
    {
      "url": "string",
      "content_type": "string",
      "file_name": "string",
      "size": numeric
    }
  ]
}
```

```

    ]
  }

```

The following table describes the parameters passed in the Text message payload request:

Parameter	Description
<i>type</i>	The static value "text".
<i>customer_id</i>	The ID of the customer to which the chatbot or CSR sends the message.
<i>message_id</i>	The unique ID of the message.
<i>csr_name</i>	The name of the CSR that is sending the message.
<i>text</i>	The text message string sent by the chatbot or CSR.
<i>attachments.[].url</i>	A temporary private URL that references an attachment file.
<i>attachments.[].content_type</i>	The content type for the file attachment.
<i>attachments.[].file_name</i>	The attachment file name.
<i>attachments.[].size</i>	The size in bytes of the file attachment.

Menu

The following is the JSON body format of the request payload that your integration layer receives from Digital Messaging Service for a menu selection:

Body (JSON)

```

{
  "type": "menu",
  "customer_id": "string",
  "message_id": "string",
  "csr_name": "string",
  "title": "string",
  "items": [
    {

```

```

        "text": "string",
        "payload": "string",
        "image_url": "string"
    }
]
}

```

The following table describes the parameters passed in the Menu payload request:

Parameter	Description
<i>menu</i>	A static value "menu".
<i>customer_id</i>	The ID of the customer to which the chatbot or CSR sends the message.
<i>message_id</i>	The unique ID of the message.
<i>csr_name</i>	The name of the CSR that is sending the message.
<i>title</i>	The title for the menu.
<i>items[].text</i>	The label for the menu option.
<i>items[].payload</i>	The postback value for the reply option.
<i>items[].image_url</i>	The URL referencing an optional image for the reply option.

Carousel

The following is the JSON body format of the request payload that your integration layer receives from Digital Messaging Service for a carousel selection:

```

Body (JSON)
{
  "type": "carousel",
  "customer_id": "string",
  "message_id": "string",
  "csr_name": "string",
  "items": [
    {
      "title": "string",
      "sub_title": "string",
      "title_image_url": "string",
      "items": [

```

```

    {
      "text": "string",
      "payload": "string",
      "description": "string",
      "image_url": "string"
    }
  ]
}
]
}

```

The following table describes the parameters passed in the Carousel payload request:

Parameter	Description
<i>type</i>	The static value "carousel".
<i>customer_id</i>	The ID of the customer to which the chatbot or CSR sends the message.
<i>message_id</i>	The unique ID of the message.
<i>csr_name</i>	The name of the CSR that is sending the message.
<i>items[].sub_title</i>	The optional subtitle for the menu.
<i>items[].title_image_url</i>	The optional title image URL for the menu.
<i>items[].items[].text</i>	The label for the menu.
<i>items[].items[].payload</i>	The postback value for the menu option.
<i>items[].items[].description</i>	The optional description for the menu option.
<i>items[].items[].image_url</i>	The URL referencing the optional image for the menu option.

Link

The following is the JSON body format of the request payload that your integration layer receives from Digital Messaging Service for a link:

```

Body (JSON)
{
  "type": "link_button",
  "customer_id": "string",
  "message_id": "string",
  "csr_name": "string",
  "title": "string",
  "label": "string",
  "url": "string"
}

```

The following table describes the parameters passed in the Link payload request:

Parameter	Description
<i>type</i>	The static value "link_button".
<i>customer_id</i>	The ID of the customer to which the chatbot or CSR sends the message.
<i>message_id</i>	The unique ID of the message
<i>csr_name</i>	The name of the CSR that is sending the message.
<i>title</i>	The title for the link button.
<i>label</i>	The label for the link button.
<i>url</i>	The URL that opens when a customer taps the button.

Typing indicator

The following is the JSON body format of the request payload that your integration layer receives from Digital Messaging Service when the CSR or chatbot is typing:

```

Body (JSON)
{
  "type": "typing_indicator",
  "customer_id": "string",
}

```

The following table describes the parameters passed in the Typing indicator payload request:

Parameter	Description
<i>type</i>	The static value "typing_indicator".
<i>customer_id</i>	The ID of the customer.

CSR ended the conversation

The following is the JSON body format of the request payload that your integration layer receives from Digital Messaging Service when the CSR ends the conversation:

```
Body (JSON)
{
  "type": "csr_end_session",
  "customer_id": "string",
}
```

The following table describes the parameters passed in the CSR ended the conversation payload request:

Parameter	Description
<i>type</i>	The static value "csr_end_session".
<i>customer_id</i>	The ID of the customer.

Delivery confirmation

On successful delivery of a request payload from your integration layer to Digital Messaging Service, the system returns a success response, 200: Successful request, for example:

```
{
  "account_id": "string",
  "source_id": "string",
  "source": "custom",
  "date": "2022-07-08T14:51:01.589Z",
  "visibility": "private",
  "recipient_profile_id": "string",
  "customer": {
    "id": "string",
    "profile_name": "string",
    "profile_id": "string"
  },
  "consumer_id": "string",
}
```

```

    "content": {
      "text": "string"
    },
    "postback": "string",
    "context_data": {
      "string": "string"
    }
  }
}

```

In a case of failure, Digital Messaging Service returns one of the following response errors:

- 400: Malformed payload (invalid message type)
- 403: Invalid/Expired JWT
- 404: API Endpoint not found
- 422: Un-processable request (missing mandatory fields, invalid JSON)

[Creating a Digital Messaging channel \(on page 113\)](#)

[Setting up the IVA for Client Channel API \(on page 153\)](#)

[Client Channel API \(on page 155\)](#)

[Client Channel API guidelines \(on page 157\)](#)

Configuring Digital Messaging channel security

To support using Pega Intelligent Virtual Assistant™ (IVA) chat on various messaging platforms, configure the security settings for your Digital Messaging channel. When you define the security settings for the applications, you can access Digital Messaging Manager for your channels in each supported production level, for example, development, staging, or production. You can then set up the supported messaging connections, for example, Web Messaging and Twilio WhatsApp for your applications.

1. Switch to Dev Studio.
2. In the header of Dev Studio, click the name of the application, and then click **Definition**.
3. In the application rule, click the **Security** tab.
4. In the **Digital Messaging security** section, in the **Manager ID** field, enter the identifier for Digital Messaging Manager.
5. In the **Manager Key** field, enter the key for Digital Messaging Manager.
6. In the **Authentication method** section, click **API Key** or **JWT**.

You will receive the configured authentication method for your system with your identifiers when Pega provisions Digital Messaging Service **Manager ID** and **Manager Key** credentials for your environment.

7. Click **Save**.
8. Switch back to App Studio.

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Digital Messaging Service overview \(on page 118\)](#)

[Digital Messaging channel overview \(on page 116\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Configuring on-premises or internal access only environments for Digital Messaging access

If your Pega Platform environment is set up on the premises or for internal access only, perform additional configuration so that your application can communicate with Digital Messaging Service. As a result, you can later create a Digital Messaging channel so that users interact with your chatbot and application on messaging platforms like Apple Messages for Business, Facebook Messenger, and WhatsApp.

For the region-specific settings described below, Pega Support will specify your region when they provision your Digital Messaging **Manager ID** and **Manager Key**. Ensure that you use the correct region configuration for your **Manager ID**. For more information, go to [Pega Support](#).

1. Allow inbound message delivery by opening the firewall for the following endpoints:
 - `https://{publicURL}/prweb/PRRestService/botAgent/v1/messaging`
 - `https://{publicURL}/prweb/PRRestService/botAgentListener/v1/accounts`

Performing this firewall configuration step ensures that your application can receive messages from Digital Messaging Service (HTTPS/443). The system rejects calls to these endpoints if you did not properly configure the **Manager ID** and **Manager Key** for your instance.



Note: Replace the `{publicURL}` string in the above endpoints with the configured public link URL for your instance.



Note: If required, also add to the list of trusted sites the dedicated Pega IP addresses that Digital Messaging Service uses when sending messages to your Pega Platform instance. Pega will specify these IP addresses when we provide your Digital Messaging Service credentials.

2. Allow outbound communication and message delivery by opening the firewall for the following endpoints for your region:



Region	Endpoints
United States (US)	• https://api.artemis.pegadigital • https://outgoing.artemis.pegadigital • https://accounts.artemis.pegadigital • https://cdn.artemis.pegadigital
Canada (CA)	• https://api.ca.um.pegadigital • https://outgoing.ca.um.pegadigital • https://accounts.ca.um.pegadigital • https://cdn.ca.um.pegadigital
Europe (EU)	• https://api.euw1.um.pegadigital • https://outgoing.euw1.um.pegadigital • https://accounts.euw1.um.pegadigital • https://cdn.euw1.um.pegadigital

Performing this firewall configuration step ensures that your application can communicate with Digital Messaging Service and send messages.

3. If you set up your Pega Platform environment behind a proxy or an API Gateway, use the proxy or gateway URL as the publicly accessible URL for your instance, and also ensure that you use a secure connection (HTTPS) for the URL.

Your proxy or API Gateway will be responsible for receiving messages and forwarding them externally to the application.

4. If Pega provisioned your Digital Messaging credentials for a region other than the United States, update the URL API settings in data transform rules for the *Data-Channel-Configuration-BotAgent* class:

- a. Switch to Dev Studio.
- b. In the navigation pane of Dev Studio, click **App**, and then search for the *Data-Channel-Configuration-BotAgent* class.
- c. Expand the **Data Model > Data Transform** section for the *Data-Channel-Configuration-BotAgent* class, and then click **pyDefault**.
- d. In the data transform rule, in the row for the **.pyManagementAPIUrl** target, in the field in the **Source** column, update the URL API endpoint for your region with the value in the first row in the table in step 2 (on page 169).
Enter the value in the field in the **Source** column in double quotation marks.
- e. Save the rule to a ruleset by clicking **Save as**.

- f. In the navigation pane of Dev Studio, click **App**, and then search for the *Data-Channel-Configuration-BotAgent* class.
- g. Expand the **Data Model > Data Transform** section for the *Data-Channel-Configuration-BotAgent* class, and then click **pySetBaseURL**.
- h. In the data transform rule, in the row for the **.pyManagementAPIUrl** target, in the field in the **Source** column, update the URL API endpoint for your region with the value in the first row in the table in step 2 (on page 169).
Enter the value in the field in the **Source** column in double quotation marks.
- i. Save the rule to a ruleset by clicking **Save as**.
- j. Switch back to App Studio.

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Configuring Digital Messaging channel security \(on page 168\)](#)

[Digital Messaging Service overview \(on page 118\)](#)

[Creating a Digital Messaging channel \(on page 113\)](#)

Template operator for IVA channel

Each conversational channel (Alexa, Digital Messaging, or Legacy Webchat) user that interacts with a Pega Platform application is mapped to a separate operator and a user account. This unique account is created automatically when a user interacts with Pega Intelligent Virtual Assistant™ (IVA) for the first time. The account is used in all subsequent interactions with the configured IVA channel.

The selected operator must at the minimum have the *PegaRULES:User4* access role, taking into consideration the application settings, for example, providing access to the application's cases. For more information, see [Standard access roles \(on page 168\)](#). Each operator that you define in the system, including a template operator, uses a language locale specified in the **Localization** section. Because each conversational channel uses a template operator, the chatbot interacts with users in one of the supported languages that you specify in the channel interface settings and in the operator locale setting.

The value selected in the **Template operator ID** list in the channel configuration specifies the name of an existing Pega Platform operator and is used as a template from which all accounts are created. Each account has a unique Alexa, Digital Messaging, or Legacy Webchat channel identifier set as its ID. You need to distinguish these types of accounts in Pega Platform so that for each Alexa, Digital Messaging, or Legacy Webchat channel instance, the account is unique. To do so, set the full name a template operator account to, for example, *Legacy Webchat Operator* and add it to the designated operator group.

Implementing an Intelligent Virtual Assistant for a conversational channel *(on page 112)*

Defining conversational channel behavior

Improve user experience in Pega Intelligent Virtual Assistant™ (IVA) by ensuring that the chatbot automatically provides meaningful and relevant responses to help users resolve an issue or address a specific business need. To speed up the business processes for your application, you can define how your chatbot behaves by automating the collection of information from users and improving the text analysis of the chat interaction. For example, when a customer requests a car insurance quote by interacting with a chatbot, the system can immediately recognize the correct subject matter, and, as a result, start an Insurance Quote business case to collect additional information about the user's car and financial history.

For relevant training materials and to test this feature on a virtual Pega instance, see the [Configuring a chatbot with Digital Messaging](#) challenge on Pega Academy.

1. Automate how the IVA interacts with users and collects information:

- Create a stand-alone conversation process to interact with users.

For more information, see [Adding a stand-alone conversation process for a conversational channel *\(on page 173\)*](#).

- Create a conversation process for a case type to automatically collect information and interact with users in a context of a case.

For more information, see [Adding a case type conversation process for a conversational channel *\(on page 179\)*](#).

2. Configure the system to create a case in a Pega Platform™ application based on user input, by adding case commands in the IVA.

For more information, see [Adding case commands for a conversational channel *\(on page 181\)*](#).

3. Configure the system to provide automatic responses to user interactions by defining response commands for the IVA.

For more information, see [Defining response commands for a conversational channel *\(on page 182\)*](#).

4. Define how the IVA analyzes user interactions so that the chatbot takes advantage of natural language processing (NLP) and adaptive analytics:

- a. Configure text analyzers for the chatbot so that the system uses natural language processing (NLP) and adaptive analytics text analysis of chat interactions.

For more information, see [Adding a text analyzer for an IVA *\(on page 188\)*](#).

- b. Define topics, which are the general subject and intent of user input that is detected by the IVA using text analysis.

For more information, see [Defining topics for text analysis for an IVA \(on page 191\)](#).



Note: To learn how to provide a seamless experience in your IVA for Digital Messaging, see [Seamless experience across Digital Messaging channels](#).

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

Adding a stand-alone conversation process for a conversational channel

To improve and simplify how Pega Intelligent Virtual Assistant™ (IVA) collects information for your application, create and configure a stand-alone conversation process. By adding a conversation process to your channel, your chatbot interacts with users in a consistent, logical, and conversational way by asking questions, displaying text, or starting another conversation process. For example, you can create a stand-alone conversation process for a chatbot that asks users their location in the United States and as a result, sends the users a message with the correct contact telephone number for your organization in the specified location.

You can also create a conversation process for a case type in your application. As a result, the system automatically responds to chat messages in the context of a case and saves user responses in the case properties. For more information, see [Adding a case type conversation process for a conversational channel \(on page 179\)](#).



Note: You can reuse a stand-alone conversation process that you save in the system in another IVA channel for your application.

1. In the Digital Messaging channel, click the **Configuration** tab.
2. In the **Conversation flows** section, click **Add conversation**.
3. In the **Create conversation** window, in the **Name** list, select **Create new conversation**.
4. In the **New conversation name** field, enter a name for the conversation process, and then click **OK**.
5. In the **Conversation command** list, enter a name for the response command for the conversation process, and then click **Submit**.
6. In the **Conversation flows** section, in the **Flow name** column, click the name of the conversation process that you created in the previous steps.

7. In the flow canvas, modify the primary path in the conversation process by connecting and configuring the following conversation shapes:

Ask a question

Asks a question to the user in the chatbot in a button, quick reply, yes/no, or open format, and saves the answer in case properties.

Send a message

Displays a text message to the user.

Conversation flow

Starts another existing conversation process within the flow that is either stand-alone or for a case type.

You can also configure additional useful shapes that are available in the context of the conversation. For more information, see [Editing the conversation smart shapes \(on page 174\)](#) and [Flow shapes \(on page 174\)](#).

8. **Optional:** To test the created flow that includes the conversation shapes in the preview console, in the top-right corner of the page, click the **Run conversation** icon.
For more information about using the preview console to simulate a conversation, see [Simulating a conversation and building a chatbot \(on page 199\)](#).
9. Click **Save**.
10. **Optional:** To display chatbot details on the **Configuration** tab, in the conversation flow toolbar, click **Go back**.

[Defining conversational channel behavior \(on page 172\)](#)

The Create stage (on page 174)

Editing the conversation smart shapes

Ensure that Pega Intelligent Virtual Assistant™ (IVA) can respond correctly by asking questions, displaying text messages, or starting another embedded conversation flow by configuring smart shapes for a conversation process. For example, for a conversation process that the system uses to dispute a cancellation fee for a requested vehicle ride, you can configure the Ask a question smart shape to ask users for the date when the fee was applied and use smart validation to ensure that the response is in the correct date format.

Instead of running in the context of the work item, you can also configure the conversation flow to run in an embedded page if you create a nested data model for your case using the object-oriented approach. For example, you can create a Change address case with two data objects of the same Address type, one specifying the customer's current address and another their registered home address. To ask users

about both their current and registered home address, you can define and reuse the Update address conversation flow that runs in an embedded page first for the current address property, and then for the registered address property.

Note:



- If your channel does not include a translation for the fields for a Send a message or Ask a question smart shape, the system will use placeholder text specified in the default language for those fields.
- If you translate the text into another language for a Send a message or Ask a question smart shape, the system creates a new language-specific ruleset only if the ruleset does not already exist for the language. If the ruleset already exists and is locked, to translate your text, ensure that you first unlock the ruleset in your application.

1. Create or edit a conversation process for a conversational channel.

For more information, see [Adding a case type conversation process for a conversational channel \(on page 179\)](#).

The system displays a flow canvas with shapes for the conversation process.

2. In the flow, right click a conversation smart shape for which you want to edit properties, and then click **View properties**:

Choices

Send a message shape

Actions

- a. If you configure multiple languages for the chatbot, in the **Configure conversation** window, select a language for the message text.
- b. In the **Message text** data field, enter the text of the message to display to the user.
- c. **Optional:** To configure additional languages for the message text, repeat steps [2.a \(on page 175\)](#) through [2.b \(on page 175\)](#).

Choices**Actions**

- d. **Optional:** To use text from an existing template, click **Use existing**.
- e. Click **Submit**.

For example, in a conversation process related to a Fee Inquiry case, you can configure the shape to display information that a charged cancellation fee for a requested vehicle ride will be refunded within the next few days.

Ask a question shape

- a. In the **Configure conversation** window, in the **Question type** list, select the type of question: **Open ended**, **Buttons**, **Quick replies**, or **Yes or No**.
- b. If you configure multiple languages for the chatbot, select a language to define for the message text, question option labels, and validation error message text.
- c. In the **Message text** data field, enter the text of the question to display to the user.
- d. **Optional:** To use text from an existing template, click **Use existing**.
- e. If you did not select an open-ended question type, in the **Options** column, enter question options in each row in the selected language, and in the **Connectors** column, select whether you want to use a connector for the option. The question options for which you do not select a connector automatically use the Else connector. For the yes or no question type, the system does not provide option connectors, only the read-only Yes and No connectors.

Choices	Actions
	f. Optional: To add additional question options, click Add item and repeat step 2.e (on page 176). g. Optional: To save a response to the question to a field, expand the Advanced section, and in the Save response to the field list, select a field in which to save the response. h. Optional: To extract an entity from an open-ended question, in the Entity to extract list, select an entity type. This field gives you the ability to find the entity in user answers and automatically validate that the answer is in the correct format, for example, date or email format. i. Optional: To repeat asking the question once, twice, three times or until a user enters a valid response, in the Number of repeats list, select 1 time, 2 times, 3 times, or Until successful. The chatbot uses smart validation and depending on the selected option for this step, the system continues to ask the question until a valid response is found. The system uses an additional connector if the chatbot does not find a response after asking the question the specified number of times. j. In the Validation error message field, enter or edit the message in the selected language to display if the answer is not valid.

Choices	Actions
	k. Optional: To configure additional languages for the message text, question option labels, and validation error message text, repeat steps 2.b (on page 176), 2.c (on page 176), 2.e (on page 176), and 2.j (on page 177). l. Click Submit. For example, in a conversation process related to a Fee Inquiry case, you can configure the shape to ask the reason for disputing the cancellation fee using the Buttons question type with three options specifying that ride did never arrive, the ride was late, and another reason. The system saves the user's response to the multiple-choice question in the <i>Dispute Reason</i> field.
Conversation flow shape	a. In the Configure flow window, in the Define flow list, select On current page, or On embedded page and then in the Select property list, select a property and a flow. To use the conversation flow in the context of the current work item, select On current page. If you use an object-oriented approach to the case data model that contains a nested structure, select On embedded page, and then select the context of which nested property to run the conversation flow. b. In the Select conversation flow list, select the name of the conversation flow to use within the conversation process. c. Click Submit.

For more information about flow shapes, see [Flow shapes \(on page 172\)](#).

3. Click **Save**.

Defining conversational channel behavior (on page 172)

The Create stage (on page 172)

Adding a case type conversation process for a conversational channel

To automate how Pega Intelligent Virtual Assistant™ (IVA) collects information for your application's cases, create and configure a conversation process for a case type. Adding a conversation process simplifies the configuration for the system and ensures that the chatbot interacts with users in a consistent, logical, and conversational way by asking questions, displaying text, or starting another conversation process. For example, you can create a conversation process for a Fee Inquiry case type designed to help users dispute an automatic cancellation charge for a vehicle that they requested in an app, when the user or driver does not show up for the requested ride.

You can also create a stand-alone conversation process that is not related to a case type directly from your conversational channel and reuse the process in other channels. For more information, see [Adding a stand-alone conversation process for a conversational channel \(on page 173\)](#).

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. In the **Case life cycle** section, in the header of the stage for the case type, click **More > Add channel process**, and then click **Digital Messaging**.

The system saves the conversational channel process as a parallel process for the stage by creating a conversation flow smart shape.

4. In the **Case life cycle** section, in the header of a parallel conversation process that contains the conversation flow, click **Configure process**.

The system displays the process flow that includes the conversation flow smart shape.

5. On the **Workflow** tab, in the conversation flow shape in the flow, click the **Add** icon.
6. In the flow canvas, modify the primary path in the conversation process by connecting and configuring the following conversation shapes:

Ask a question

Asks a question to the user in the chatbot in a button, quick reply, yes/no, or open format, and saves the answer in case properties.



Send a message

Displays a text message to the user.

Conversation flow

Starts another existing conversation process that is stand-alone or for a case type.

You can also configure additional useful shapes that are available in the context of the conversation. For more information, see [Editing the conversation smart shapes \(on page 174\)](#) and Flow shapes (on page).

7. Click **Save**.

The system saves your configured conversation process for the chatbot as a conversation flow in the parallel process for a case.

8. **Optional:** To display the case life cycle view, click **Stages & steps**.

[Defining conversational channel behavior \(on page 172\)](#)

The Create stage (on page)

Adding a conversational channel to a case type process

To improve the accuracy of Pega Intelligent Virtual Assistant™ (IVA) responses to a case, add a conversational channel to a process for a case type. You can also define the conditions that control when the conversation begins in the IVA. For example, you can add an IVA for Digital Messaging to the car insurance case type. The system creates a parallel process for this IVA channel and, as a result, you can later define a set of simple questions to collect information for a car insurance case.

1. In the navigation pane of Dev Studio, click **Case types**, and then click the case type that you want to open.
2. In the navigation pane of App Studio, click **Case types**, and then click the case type that you want to open.
3. On the **Workflow** tab, click **Life cycle**.
4. Click the stage to which you want to add a conversational channel:
 - To add a conversational channel to the Create stage, click **More > Add create process > Channel name**.
 - To add a conversational channel to any other stage, click **More > Add channel process > Channel name**.

A parallel process for the IVA channel is added to the stage.

5. **Optional:** To define the conditions that control when the conversation begins, update the process settings:

- a. On the **Workflow** tab, click the parallel process for the IVA channel.
 - b. On the **General** tab of the property panel, in the **Start Process** list, select **When Rule**.
 - c. Select **Custom condition**, and then define the condition that determines when the conversation in the IVA starts.
 - d. In the field below the **Start Process** list, enter or select a when rule that determines when the conversation in the IVA starts.
6. Click **Save**.

At run time, the system asks a series of questions to collect information about a Pega Platform case.

[Adding a case type conversation process for a conversational channel \(on page 179\)](#)

The Create stage (on page)

Adding case commands for a conversational channel

By defining case commands in the system, you ensure that Pega Intelligent Virtual Assistant™ (IVA) can create top-level business cases based on detected user input. As a result, you improve the completion of business processes that are requested by users in your application. Based on the analysis of a text message or a voice command, the chatbot matches the best case command for a topic, the detected subject matter, and then automatically creates a corresponding business case. The chatbot can also automatically extract entities from user input, and then copy them to case properties. For example, when a customer requests a car insurance price quote in an interaction with the chatbot, the system automatically creates a Car Insurance case and asks follow-up questions about the customer's car and credit history.

All case commands and response commands that you define are topics.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Configuration** tab.
5. In the **Content** section, select a method for modifying a case command:
 - To add a new case command, click **Add case type**.
 - To edit an existing case command, click the **Switch to edit mode** icon for a case type.
6. In the **Case type** list in the **Create command** window, select a case type that you want to create when the user enters a specific command.
7. In the **Create case command** field, enter text that triggers the system to create the case.

Through text analysis, a user command provides the input that determines the best case command type to create by the IVA.

8. If the case command requires authentication, select the **Requires authentication** checkbox.
9. **Optional:** To add more case commands, repeat steps [5 \(on page 181\)](#) through [8 \(on page 182\)](#).
10. Click **Submit**.
11. Click **Save**.

[Defining conversational channel behavior \(on page 172\)](#)

Defining response commands for a conversational channel

To automate business processes and save the time that case workers spend answering simple questions that are typical of a case type, you can define response commands so that Pega Intelligent Virtual Assistant™ (IVA) correctly reacts to user input. The chatbot detects the best response for a topic, which is the detected subject matter of the conversation, by using text analysis, and then automatically invokes a response type. For example, when a customer asks for help during an interaction with your chatbot, the system can immediately display a menu of available options as the response command.

You can also add case commands to the IVA, so that the system automatically starts a case in the application based on user input. All suggested cases and suggested responses that you define are topics.

Note:



- If your channel does not include a translation for the response command fields, then the system will use the placeholder text specified in the default language for those fields.
- If you translate text into another language for a response command, the system creates a new language-specific ruleset only if the ruleset does not already exist for the language. If the ruleset already exists and is locked, to translate your text, ensure that you first unlock the ruleset in your application.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the channel, click the **Configuration** tab.
5. In the **Content** section, select a method for modifying a response command:

- To add a new response command, click **Add response**.
- To edit an existing response command, click the **Switch to edit mode** icon for a response displayed in the list.

6. In the **Response type** list in the **Response configuration** window, select and configure a response type:

Choices	Actions
Authenticate the user in the system	- Select Authenticate. - If you configure multiple languages for the chatbot, select a language for the response text. - In the Response text area, enter the text that you want to display in the selected language. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 183) through 6.c (on page 183). Optional: To add another text to display to the user, below the Response text area, click Add new, and enter additional text.
Cancel the current work in the interaction session	- Select Cancel. - If you configure multiple languages for the chatbot, select a language for the response text. - In the Response text area, enter the text that you want to display in the selected language. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 183) through 6.c (on page 183). Optional: To add another text to display to the user, below the Response text area, click Add new, and enter additional text.

Choices	Actions
	f. Optional: To add a command to a menu that the system displays after you cancel current work, click Add item. g. In the Menu item list, select the command that you want to invoke for the menu item. h. In the Label field, enter the menu item name that you want to display in the system. i. Optional: To add more commands that you want to display as menu items, repeat the last three steps.
Display a rich text menu by using a data transform in the IVA for Legacy Webchat	a. Select Custom. b. In the Data transform field, select a data transform rule. The rich text menu in the IVA for Legacy Webchat looks similar to a Facebook generic template. The menu consists of a title, subtitle, link to an image, and up to three menu options. You can use the default <i>pyPopulateOutboundData</i> data transform as an example for the IVA.
Link to a specified Facebook account in the IVA for Facebook	a. Select Link. b. If you configure multiple languages for the chatbot, select a language for the response text. c. In the Response text area, enter the text that you want to display to the user in the selected language. d. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 184) through 6.c (on page 184). e. Optional: To add another text to display to the user, below the Response

Choices	Actions
	text area, click Add new, and enter additional text. f. In the Title field, enter a title for the Facebook account to which you want to link. g. In the Authentication callback URL field, enter the authentication URL for the Facebook account to which you want to link.
Display a menu with a list of available commands	a. Select Menu. b. In the Menu template list, select to display the menu items as simple menu buttons or quick replies. c. If you configure multiple languages for the chatbot, select a language to define for the response text and menu item labels. d. In the Response text area, enter the text that you want to display to the user in the selected language. e. Optional: To add another text to display to the user, below the Response text area, click Add new, and enter additional text. f. Optional: To add a command to a menu, click Add item. g. In the Menu item list, select the command that you want to invoke for the menu item. h. In the Label field, enter the menu item name that you want to display in the system in the selected language. i. To add more commands to display as a menu item, repeat the last three steps. j. Optional: To specify response text and menu item labels in additional lan-

Choices	Actions
Restart work on a case that is currently being processed in the session	languages, repeat steps 6.c (on page 185), 6.d (on page 185), and 6.h (on page 185). - Select Restart. - If you configure multiple languages for the chatbot, select a language for the response text. - In the Response text area, enter the text that you want to display to the user in the selected language. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 186) through 6.c (on page 186). Optional: To add another text to display to the user, below the Response text area, click Add new, and enter additional text.
Display a text message to the user	- Select Text. - If you configure multiple languages for the chatbot, select a language for the response text. - In the Response text area, enter the text that you want to display to the user in the selected language. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 186) through 6.c (on page 186). Optional: To add another text to display to the user, below the Response text area, click Add new, and enter additional text.

7. In the **Response command** field, enter or modify a user command for the response type. Through text analysis, a user command provides the input that determines the best response command type that the system detects.

8. If the response command requires authentication, select the **Requires authentication** checkbox.
9. Click **Submit**.
10. **Optional:** To add or edit more response commands, repeat steps 5 (on page 182) through 9 (on page 187).
11. Click **Save**.

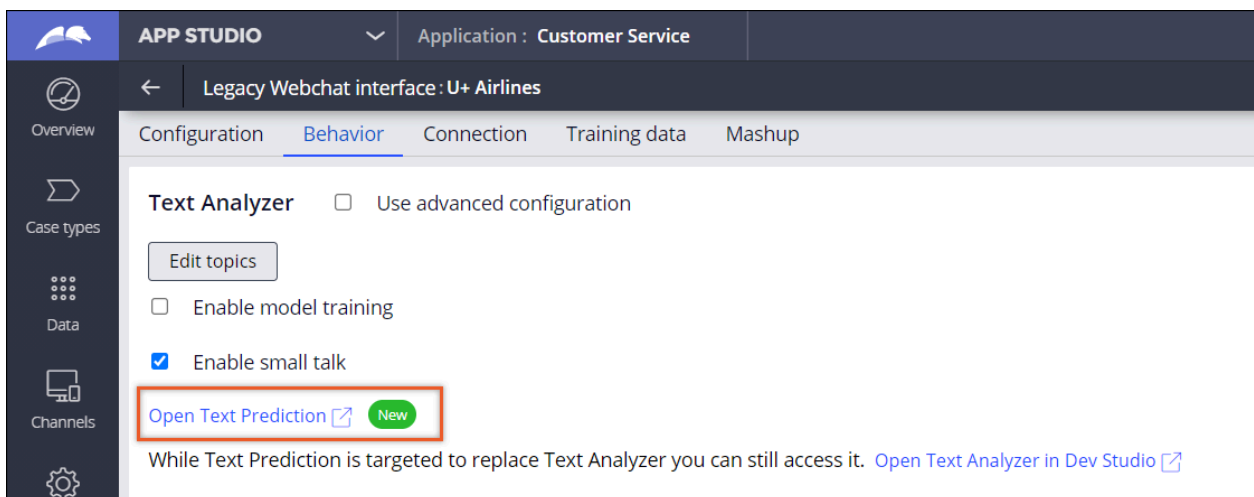
[Defining conversational channel behavior \(on page 172\)](#)

Configuring a text prediction for an IVA

To enhance the artificial intelligence of Pega Intelligent Virtual Assistant™ (IVA) so that the system provides better responses and detects entities and the topic in the messages, you can configure a text prediction for an application.

Tip: Text predictions are targeted to replace text analyzers. As a best practice, configure text analytics for your conversational channels by using text predictions instead of text analyzers. For more information, see [Text analysis concepts \(on page 21\)](#).

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon for your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.
5. In the **Text Analyzer** section, click **Open text prediction**.



Opening a text prediction from a channel

6. In Prediction Studio, configure the text prediction for the chatbot.

For more information, see [Analyzing messages with text predictions \(on page 188\)](#).

Related information

[Adding a text analyzer for an IVA \(on page 188\)](#)

[Text analysis concepts \(on page 21\)](#)

Adding a text analyzer for an IVA

To enhance the artificial intelligence of Pega Intelligent Virtual Assistant™ (IVA) so that the system provides better responses and detects entities and the topic, the subject matter, you can configure one or more advanced text analyzers together for an application. In that case, the text analyzers examine user input one by one, until the system finds a response and detects the correct data.

You can also configure one text analyzer to run only within the case context, and another one to run outside of the case context. For example, to refine interaction with a user when a case is started in the IVA, you can specify the iNLP advanced text analyzer to improve the text analysis of user input, while using the simple exact match text analyzer if the system does not start a case during a chat session.

Tip: Text predictions are targeted to replace text analyzers. As a best practice, configure text analytics for your conversational channels by using text predictions instead of text analyzers. For more information, see [Text analysis concepts \(on page 21\)](#).

Note: The system uses natural language processing (NLP) and adaptive analytics text analysis to detect topics and entities in the interaction conversation.

Note: To learn more about troubleshooting natural language processing (NLP) issues in your chatbots, see also [Troubleshooting NLP](#).

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon for your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.

5. In the **Text Analyzer** section, select the **Use advanced configuration** checkbox.
6. In the **Text Analyzer** section, select a method for configuring a text analyzer:
 - To create a text analyzer, click **Add text analyzer**.
 - To edit an existing text analyzer, click the **Switch to edit mode** icon next to the text analyzer that you want to edit.
7. In the **Text Analyzer type** list, select and configure a text analyzer:

Choices	Actions
Exact match	Configure the default text analyzer that exactly matches user input to a response. Select whether to use the text analyzer within the case context, outside of the case context, or both.
Pega NLP	Configure an advanced text analyzer that uses the best approximate match by using advanced NLP and artificial intelligence: a. Select or define a text analyzer rule for this definition type with the sentiment, classification, topic, and entity extraction analysis. b. Select whether you want to detect entities or topics within the case context, outside of the case context, or both. c. To ensure that the chatbot engages users in casual conversations, select the Enable small talk checkbox. d. In the Text analyzer rule field, create or select a text analyzer rule. e. To modify the text analyzer rule settings, click the Open Text Prediction field. For more information, see Building text analyzers (on page).

Choices	Actions
	f. To modify the text analyzer rule settings, click Open text analyzer in Dev Studio, make changes, and then in the lower-left of the page click Back to Pega App Studio. The option to modify the text analyzer rule is only available if you have access to Dev Studio. For more information, see Building text analyzers (<i>on page</i>).
iNLP  Note: The iNLP text analyzer is associated with the default text prediction for the channel.	Configure an advanced intelligent NLP text analyzer / text prediction that uses adaptive analytics text analysis: - Select whether you want to detect entities or topics within the case context, outside of the case context, or both. - To allow users to add training data for the text models, select the Enable model training checkbox. - To ensure that the chatbot engages users in casual conversations, select the Enable small talk checkbox. - To modify the text prediction settings, click Open Text Prediction. For more information, see Analyzing messages with text predictions (<i>on page</i>). - To modify the text analyzer rule that underlies the text prediction, click Open Text Analyzer in Dev Studio, make changes, and then in the lower-left of

Choices	Actions
	the page, click Back to Pega App Studio. For more information, see Building text analyzers (<i>on page</i>). This type of analysis integrates text analytics with strategies, propositions, and interaction history to provide the context for making better next-best-action decisions. For more information, see Customizable Interaction API for text analytics (<i>on page</i>).

8. Click **Submit**.
9. **Optional:** To add or configure more text analyzers for the IVA, repeat steps 6 (*on page 189*) through 8 (*on page 191*).
10. Click **Save**.

[Exploring text predictions and text analyzers \(*on page 23*\)](#)

[Text analysis concepts \(*on page 21*\)](#)

[Defining conversational channel behavior \(*on page 172*\)](#)

Defining topics for text analysis for an IVA

Define topics in Pega Intelligent Virtual Assistant™ (IVA) so that the system can adjust responses by analyzing user input through natural language processing (NLP) and text analytics models. The topics represent the general subject matter of the user input that the system detects to help the chatbot select the correct response commands, case commands, and conversation processes during a chat interaction. For example, for a travel agency application, when a customer sends a message to the chatbot to book a flight, the system correctly detects the subject matter of the chat conversation and starts a Book Flight business case.

If you configure your chatbot for multiple languages, you do not translate topic and entity names into other languages because they are internal application values.

Note: All suggested cases that you define in the system are topics.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.
5. To enable the advanced configuration mode, in the **Text Analyzer** section, select the **Use advanced configuration** checkbox.
In the advanced configuration mode, you configure the advanced text analyzers in the system that use NLP and adaptive analytics. If you do not use advanced configuration, you configure the default text analyzer that exactly matches user input to a response. For more information, see [Exploring text predictions and text analyzers \(on page 23\)](#).
6. In the **Text Analyzer** section, click **Edit topics** for the text analyzer that you want to edit.
7. In the **Edit topics** window, click **Add item**.
A topic is the general subject, the intent of a voice command, or a text message that the system detects using text analysis.
8. If you configure multiple languages for the chatbot, select a language.
If your channel does not include a translation for the fields in the selected language, then the system will use placeholder text specified in the default language for those fields.



Note: If you translate text into another language, the system creates a new language-specific ruleset only if the ruleset does not exist for the language already. If the ruleset already exists and is locked, to translate your text, ensure that you first unlock the ruleset in your application.

9. In the table columns, specify a topic and matching criteria for the selected language:
 - In the **Topic** field, enter the natural language processing topic for the IVA to use as the response. The NLP text analyzer determines the topic based on criteria that either matches exactly, matches approximately, and/or never matches the user input.
 - In the **Approximate match** field, to configure the IVA to look for approximate matches in the user input, enter the words and phrases that you want to use, and then separate them with commas. For example, for an initial greeting responses, enter: hello, good morning.

- In the **Must match** field, to configure the IVA to look for specific items in the conversation content, enter the words and phrases that you want to exactly match in the topic, and separate them with commas. For example, for an initial greeting response, enter hi.
 - In the **Never match** field, enter comma-separated words or phrases to always ignore when matching for the topic. For example, for an initial greeting response to never match a goodbye greeting, enter bye.
10. **Optional:** To configure additional languages for the topic match texts, repeat steps [8 \(on page 192\)](#) through [9 \(on page 192\)](#).
 11. **Optional:** To add more topics to a text analyzer, repeat step [7 \(on page 192\)](#) and [10 \(on page 193\)](#).
 12. Click **Submit**.
 13. Click **Save**.

The IVA uses text analysis to match user input to a topic, a suggested case or a suggested response.

[Adding a text analyzer for an IVA \(on page 188\)](#)

[Text analysis concepts \(on page 21\)](#)

[Defining conversational channel behavior \(on page 172\)](#)

Configuring system responses for a conversational channel

A conversational channel contains several predefined system responses for standard events that happen during an interaction between a user and Pega Intelligent Virtual Assistant™ (IVA). To fit the needs of your organization, you can customize system responses so that an IVA is more responsive to user input. For example, you can modify the initial system response, the voice command, or the text message that the IVA invokes when the user starts a new interaction session.

To ensure that the IVA responds correctly, apart from modifying the system responses, you can define response commands when you configure the conversational channel behavior. For more information, see [Defining response commands for a conversational channel \(on page 182\)](#).

 **Note:**



- If your channel does not include a translation for the system response fields, then the system will use placeholder text specified in the default language for those fields.
- If you translate response text into another language for a system response, the system creates a new language-specific ruleset only if the ruleset does not already exist for the language. If the ruleset already exists and is locked, to translate your text, ensure that you first unlock the ruleset in your application.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Alexa, Digital Messaging channel.
4. In the channel, click the **Behavior** tab.
5. In the **System Responses** section, click the **Switch to edit more** icon for one of the system responses:
 - To display a welcome message to the user when they start an interaction with the IVA, edit the **Welcome** response.
 - To display a warning that the user input does not contain a matching command, edit the **No match** response.
 - To display a message stating that multiple commands match the user input, edit the **Multiple matches** response. This response is not available in the IVA for Alexa.
 - To display a message that a system error occurred for the IVA, edit the **Internal error** response.
 - To display an authentication prompt to log in to a Pega Platform application, edit the **Authentication** response. This response is not available in the IVA for Alexa.
 - To display a success message immediately after a user successfully authenticates, edit the **Authentication-Success** response. This response is not available in the IVA for Alexa.
 - To display a failure message immediately after a user fails to authenticate, edit the **Authentication-Fail** response. This response is not available in the IVA for Alexa.
6. In the **Response type** field, select and configure a response type:

Choices

Authenticate the user in the system

Actions

- a. Select **Authenticate**.
- b. If you configure multiple languages for the chatbot, select a language for the response text.

Choices	Actions
	c. In the Response text area, enter the text to display when the user authenticates in the system. d. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 194) through 6.c (on page 195).
Cancel the current work in the interaction session	a. Select Cancel b. If you configure multiple languages for the chatbot, select a language for the response text. c. In the Response text area, enter the text that you want to display in the selected language. d. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 195) through 6.c (on page 195). e. Optional: To add a command to a menu you display after you cancel current work, click Add response. f. In the Menu item list, select a command that you want to invoke for the menu item. g. In the Label field, enter a menu item name that you want to display in the system. h. To add more commands that you want to display as menu items, repeat steps 6.e (on page 195) through 6.g (on page 195).
Start a top-level case	a. Select Create work. b. In the Case type list, select a case type to open.

Choices	Actions
Display a rich text menu for the IVA for Legacy Webchat	a. Select Custom. b. In the Data transform field, select or configure a data transform rule that is used to display a rich text menu.
Link to a Facebook account for the IVA for Facebook	a. Select Link. b. In the Title field, enter a title for a Facebook account that you want to link to. c. In the Authentication callback URL field, enter the authentication callback URL for a Facebook account that you want to link to.
Display a menu with a list of available commands	a. Select Menu. b. If you configure multiple languages for the chatbot, select a language for the response text. c. In the Response text area, enter the text that you want to display to the user in the selected language. d. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 196) through 6.c (on page 196). e. Optional: Add a command to a menu by clicking Add response. f. In the Menu item list, select the command that you want to invoke for the menu item. g. In the Label field, enter the menu item name that you want to display in the system. h. To add more commands to display as a menu item, repeat steps 6.e (on page 196) through 6.g (on page 196).

Choices	Actions
Restart work on a case that is currently being processed in the session	a. Select Restart. b. If you configure multiple languages for the chatbot, select a language for the response text. c. In the Response text area, enter the text that you want to display to the user in the selected language. d. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 197) through 6.c (on page 197).
Display a text message to the user	a. Select Text. b. If you configure multiple languages for the chatbot, select a language for the response text. c. In the Response text area, enter the text that you want to display to the user in the selected language. d. Optional: To configure additional languages for the response text, repeat steps 6.b (on page 197) through 6.c (on page 197).

7. **Optional:** To save the text from the **Response text** area in a template, click **Add existing**.
8. Click **Submit**.
9. Click **Save**.

[Defining conversational channel behavior \(on page 172\)](#)

Guidelines for sharing text analytics models across channels

Develop and maintain Pega Intelligent Virtual Assistant™ (IVA) for your application more efficiently by reusing text analytics models in multiple IVAs and email bots. Sharing text analytics models makes channel behavior consistent in all of the IVAs and email bots, and decreases the development time for conversational channels. Reusing text analytics models is an advanced configuration feature for IVAs. For example, you can share the topic model in multiple text analyzers for IVAs, so that the system consistently detects the correct subject matter in conversation text for each IVA.



Channels in your applications, such as IVAs or email bots, work closely with text analyzers. Pega Platform provides you with several out-of-the-box text analytics models for topic, small talk, and sentiment detection. The models are referenced in a text analyzer rule for each channel. When you create an IVA or an Email channel in your application, the system saves a unique instance of the text analyzer and its related text analytics models in the child hierarchy of the *Data-Decision-Request* base class. The topic model referenced by the *pyInteractionTaxonomy* rule is saved in the following classes:

- *Data-Decision-Request-Email-<UNIQUE_ID>* class for email bots
- *Data-Decision-Request-MCP-<IVA_TYPE>-<UNIQUE_ID>* class for IVAs

Note: You can share and modify text analytics models in Dev Studio.

Guidelines

Follow these guidelines when sharing text analytics models across channels:

- Create instances of text analytics model rules in the parent *Data-Decision-Request* base class, instead of within the child hierarchy of this class.
- Select the rule that you save in the *Data-Decision-Request* base class for each text analyzer rule of an IVA and email bot that you want to share a model.
- Ensure that there are no instances of this rule defined at the child class level. Otherwise, during rule resolution the system will give preference to the rule at the child class level instead of the one at the parent class level.
- If you want to reuse the models for only a subset of the IVAs and email bots, to avoid confusion about which version of the model the system uses, save the model rule at the parent *Data-Decision-Request* base class level with a different name from the default name. For example, for the topic model, save the *pyInteractionTaxonomy* rule at the parent class level with another name.
- The small talk model specified by the *pySmallTalk* rule for a text analyzer is only available for IVAs. This rule is available in the *@baseclass* base class, which means that any IVA channel can access it. You can only share this model rule in IVAs, not email bots.
- Do not share instances of text analyzer rules between multiple IVAs and email bots. You can only share the text analytics models referenced by a text analyzer across different channels.

Example

When you create an Email channel, the system saves an instance of the topic model rule for the email bot in the *pyInteractionTaxonomy* rule in the *Data-Decision-Request-Email-EDVVSXTJ* class, under **Decision > Decision Data**. When you create a Legacy Webchat channel, the system saves an instance of the topic model for the IVA in the *pyInteractionTaxonomy* rule, in the *Data-Decision-Request-MCP-WebChat-KKYYUFTN*

class, under **Decision > Decision Data**. To share a text analytics model across multiple channels, for example, the topic model *pyInteractionTaxonomy* rule, save an instance of this rule with another name in a parent *Data-Decision-Request* base class, under **Decision > Decision Data**. To share the topic model, you then select the instance of the *pyInteractionTaxonomy* rule from this location for each text analyzer rule of an IVA or an email bot.

[Adding a text analyzer for an IVA \(on page 188\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

Simulating a conversation and building a chatbot

Before moving Pega Intelligent Virtual Assistant™ (IVA) to a production environment, you can verify whether the system works correctly by interacting with the chatbot using the preview console. You use the preview console to simulate a conversation for your business application and build the chatbot, to ensure that the chatbot knows how to correctly respond and detect information from user input. Working in the preview console also improves the text analysis and artificial intelligence of the IVA channel. After making any changes to the channel configuration in the preview console, rebuild the text analytics model.

1. Verify the detected information by the chatbot.
For more information, see [Verifying detected topics and entities \(on page 199\)](#).
2. Update the chatbot questions and response text.
For more information, see [Modifying chatbot response questions \(on page 201\)](#).
3. Extract data from a conversation.
For more information, see [Creating entities for an IVA \(on page 202\)](#) and [Mapping entities in conversation text \(on page 203\)](#).
4. Apply any simulation changes to the text analytics model.
For more information, see [Applying simulation changes to a text analytics model \(on page 204\)](#).

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

Verifying detected topics and entities

To ensure that Pega Intelligent Virtual Assistant™ (IVA) responds correctly to user requests during a chat session, you can first simulate conversations with the chatbot and then verify the detected information in the preview console for a conversational channel. As a result, the system becomes more responsive and improves its text analysis and artificial intelligence. For example, you can verify that the system correctly detects the topic related to car insurance, the subject matter, and detects a car model and make, when a user asks for a car insurance quote for their car in the chat session.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.
5. If you configure multiple languages for the chatbot, at the top-right of the preview console, select a language.
6. Turn on the **Show analysis** switch.

When you enter sample text in the preview console, the analysis shows the topic detected and the system's confidence in the correctness of the detected topic, as a percentage.

7. In the preview console in the right panel, type in a typical user phrase that starts a business case.

The preview console displays the detected topic below the text that you enter, and starts the associated business case in the chatbot.

8. Correct the topic (subject matter) or entity detected in user input:

Choices

If no topic was found

Actions

- a. Click **Define topic** below the entered text.
- b. In the **Topic** field, press the Down arrow key, and then select the correct topic.
- c. In the field below the topic name, enter a sample of user input to associate with the topic.
- d. **Optional:** To add additional sample phrases to associate with the topic, click **Add example**, and then in the field below, enter the text.
- e. Click **Submit**.

If the wrong topic was found

- a. Click the topic name located below the entered text.
- b. In the **Topic** field, press the Down arrow key, and then select the correct topic.

Choices	Actions
	c. In the field below the topic name, enter a sample of user input to associate with the topic. d. Optional: To add additional sample phrases to associate with the topic, click Add example, and then in the field below, enter the text. e. Click Submit.
If no entity was found	a. In the text section below the detected sentiment, highlight and right-click the text that you want to map to the new entity. b. Click New entity. c. In the Create new entity window, in the Entity name field, enter a name for the entity, and then click Submit.
If the wrong entity was found	a. In the text section below the detected sentiment, highlight and right-click the text for an existing entity. b. Click the name of another entity.

9. Click the **Like** icon next to the topic name located below the entered text.

Clicking this icon is important in the IVA training process, and improves the correct detection of topics.

10. Click **Save**.

[Simulating a conversation and building a chatbot \(on page 199\)](#)

Modifying chatbot response questions

Ensure that Pega Intelligent Virtual Assistant™ (IVA) asks the correct follow-up questions to user input, by modifying the chatbot responses. By simulating a conversation and modifying chatbot response questions in the preview console, the IVA learns how to obtain additional clarifying information about the correct case type from the user in a quick and conversational manner. For example, in a request for a car insurance quote, during a user interaction, an IVA asks questions about the make, model, and year of production of the car, and then automatically stores the answers in the properties of the business case.



1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.
5. If you configure multiple languages for the chatbot, at the top-right of the preview console, select a language.
6. Turn on the **Show analysis** switch.
7. In the preview console, in the right panel, enter a command.

The preview console starts a *Car Insurance* case and displays a follow-up question, for example: What is the make of the foreign-made car?

8. To view additional options for the displayed question, click the **View folder contents** icon below the question in the preview console.
9. In the **Edit question** window, in the **Question text** field, edit the text for the question.
10. **Optional:** To map the follow-up question from the IVA to a property, edit the **Map answer to property** field.
11. Click **Save**.

[Simulating a conversation and building a chatbot \(on page 199\)](#)

Creating entities for an IVA

Ensure that Pega Intelligent Virtual Assistant™ (IVA) detects the correct information in chat sessions by creating entities. Entities enable the IVA to extract pieces of data from within a chat message, such as a location, date, or postal code. For example, if you want the system to detect vehicle makes in user input, you can create the *CarMake* entity. At run time, your IVA then uses the detected information and maps it to case type properties, to provide the correct response to user requests and save time for the users handling the case.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.
5. If you configure multiple languages for the chatbot, at the top-right of the preview console, select a language.
6. Turn on the **Show analysis** switch.



7. In the preview console, in the right panel input field, enter a sample question.

The analysis in the preview console shows the sentiment of the user input (positive, negative, neutral), the detected topic, the confidence in the detected topic (as a percentage), and the detected entities.

8. In the displayed text for the question that you entered in the previous step, in the create case section, highlight and right-click the text that you want to map to the new entity, and then click **New entity**.
9. In the **Create new entity** window, in the **Entity name** field, enter a name for the entity, and then click **Submit**.
10. To create more entities for detection in user input, repeat steps [8 \(on page 203\)](#) through [9 \(on page 203\)](#).
11. Click **Save**.

[Mapping entities in conversation text \(on page 203\)](#)

[Simulating a conversation and building a chatbot \(on page 199\)](#)

Mapping entities in conversation text

To prevent your Pega Intelligent Virtual Assistant™ (IVA) from asking for information that the user has already provided, you can optionally configure entity mapping definitions for the cases configured in your channel. When the IVA creates a case during a chat session, the system applies the associated entity mapping and copies any entities extracted from the user input into the case properties. For example, for a *Car Insurance* case, you can map the *CarModel* and *CarMake* entities to the *.CarModel* and *.CarMake* case properties. When these entities are detected in a chat conversation, their values are automatically assigned to the case properties and the system does not ask the user about the car make and model.

When a case property has a value stored in it that was mapped from a detected entity, the IVA will skip over this question shape. This prevents the chatbot from asking for information that the user has already provided.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel that you select, click the **Behavior** tab.
5. At the top of the preview console, turn on the **Show analysis** switch.
6. In the preview console, in the right panel input field, enter a sample question.

The analysis in the preview console shows the sentiment of the user input (positive, negative, neutral), the detected topic, the confidence in the detected topic (as a percentage), and the detected entities.

7. In the create case section, click the **Open response configuration** icon.
8. In the **Response configuration** window, click the **Entities extraction** tab.
9. Add an entity to a case property mapping by clicking **Add mapping**.
10. In the **Entity** list, click an entity.
11. In the **Case property** list, click a case property that you want to map to the entity.
12. To add another entity to the case property mapping, repeat steps [9 \(on page 204\)](#) through [11 \(on page 204\)](#).
13. Click **Submit**.

[Creating entities for an IVA \(on page 202\)](#)

[Simulating a conversation and building a chatbot \(on page 199\)](#)

Applying simulation changes to a text analytics model

After you make corrections to natural language processing (NLP) records from the preview console relating to topic, entities and language for Pega Intelligent Virtual Assistant™ (IVA), apply the changes to the text analytics model. Rebuilding the text analytics model with additional or modified records improves the IVA's ability to understand user requests, and improves how the system responds to detected topic, entities, and language in the user input. When you update a detected topic or add entities during a chat simulation in the preview console, the system saves the changes as a training record.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.

The number of records that you previously modified is displayed at the top of the preview console.

5. At the top of the preview console, click **Build model**.

[Simulating a conversation and building a chatbot \(on page 199\)](#)

Training the model for the IVA channel

To enable Pega Intelligent Virtual Assistant™ (IVA) in your application to react to user requests, teach the system to recognize different types of input in a chat conversation. When you train the text analytics model for the IVA, the system learns from training records, improves the artificial intelligence algorithms, and learns to react in the correct way. For example, after you train the text analytics model for an IVA and a user asks about a car insurance quote in the chatbot, the system detects the correct topic and entities. As a result, the IVA starts a car insurance business case and asks the user several follow-up questions about their car and credit history.

To ensure that the IVA learns from the training records and detects the correct topics and entities, apply the training changes to the text analytics model for the system. You apply changes to the model after you enable the training data recording, and then create and review the training records.

There are several guidelines to consider when you train the text analytics model for your chatbot. The approach that you take when training model-based entities is different than when you are building rule-based entities. By observing the guidelines for training text analytics models, you decrease the time it takes to train an IVA by improving the accuracy of the model, so that the system can more efficiently detect correct, relevant information in a conversation. For more information, see [Best practices when training model-based entities in an IVA \(on page 211\)](#), [Best practices when building rule-based entities in an IVA \(on page 212\)](#), and [Best practices for cleaning up training data in an IVA \(on page 213\)](#).



Note: To learn how to improve the user experience in the IVA for Digital Messaging, see [Improving the user experience of chatbots](#).

1. Enable training of your text analytics model for the IVA.
For more information, see [Enabling training data recording for an IVA \(on page 206\)](#).
2. Improve the IVA model by providing sample training records.
For more information, see [Creating training data manually for an IVA \(on page 207\)](#).
3. Update the training data by ensuring that the IVA detects the correct information by using text analysis.
For more information, see [Correcting training data in an IVA \(on page 207\)](#).
4. **Optional:** To copy the training data to another IVA, export and import the training records.
This action makes the IVA easier to maintain and build in production, QA, and development environments. For more information, see [Transferring training data to another IVA \(on page 209\)](#).
5. Make the IVA learn from the training data by building a text analytics model.
This action improves the artificial intelligence of the system. For more information, see [Applying changes to a text analytics model for an IVA \(on page 211\)](#).

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

[Text analysis concepts \(on page 21\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

Enabling training data recording for an IVA

To help Pega Intelligent Virtual Assistant™ (IVA) make better predictions when responding to users, enable the recording of training data in the system. You can then modify and classify the training data by reviewing the data and applying changes to the text analytics model. As a result, you ensure that the IVA correctly detects the topics and entities in chat conversations with users.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Alexa channel.
4. In the channel, click the **Behavior** tab.
5. In the **Text Analyzer** section, activate the recording of training data:

Choices	Action
For the default text analyzer	a. Select the Enable model training checkbox.
For advanced configuration	a. Select the Use advanced configuration checkbox. b. In the text analyzer list, click the Switch to edit mode icon for a text analyzer. c. In the Text analyzer configuration window, select the Enable model training checkbox. d. Click Submit .

6. Click **Save**.

[Training the model for the IVA channel \(on page 205\)](#)

[Text analysis concepts \(on page 21\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

Creating training data manually for an IVA

Ensure that Pega Intelligent Virtual Assistant™ (IVA) interprets chat messages from users in the correct way by manually creating training records from which the system can learn. Training records provide valuable information for the IVA that strengthens the artificial intelligence algorithms and improves the accuracy of your text analytics model. By training the model, you ensure that the system correctly detects topics and entities, the subject matter and other vital information extracted from the chat conversation. For example, you can add a training record in which customers request bank loans. If you connect that record to a bank loan topic, the system learns to associate this information with bank loan cases.

You review the training data before applying the changes to the text analytics model.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the channel, click the **Training data** tab.
5. In the toolbar, click **Add records**.
6. In the **Topic** field in the **Create new training record** window, press the Down arrow key, and then select the topic to which the training record relates, for example, **Bank loan**.
7. If you configure multiple languages for the chatbot, select a language for the training record.
8. In the text field below the **Topic** field, enter the text for the training record.
The text must be a good representation of sample user input, from which the IVA can learn to respond to messages.
9. Click **Create record**.

The system displays the training record in the list on the **Training data** tab. The **Review training data** section displays the content of the training record with the detected entities highlighted. The **NLP analysis** section displays the detected topic and entity types in the training data.

10. **Optional:** To add more training records for the IVA, repeat steps 5 ([on page 207](#)) through 9 ([on page 207](#)).
11. Click **Save**.

[Training the model for the IVA channel \(on page 205\)](#)

Correcting training data in an IVA

Improve the artificial intelligence of Pega Intelligent Virtual Assistant™ (IVA) and teach the system to more accurately respond during chat conversations by correcting the IVA training data and rebuilding the text



analytics model. When the IVA correctly detects the subject matter (topic) and additional key information (entities) in a conversation, the system can then escalate the reported issue, start a related business case and ask the user for more information. For example, after identifying the relevant topic and entities in a message, an IVA detects a request for a bank loan. The system then automatically creates a car loan business case and asks the user follow-up questions about their personal information, income, and financial history.

When you enable the recording of training data and the IVA begins a conversation with a user, the system saves each chat message as a training record. You can also manually create a training record that contains information from sample conversations with users. Once you correct the data detected in the training record, you mark the record as reviewed. The system then uses the reviewed training data when you rebuild the improved text analytics model.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the channel, click the **Training data** tab.
5. In the list of training records, select a data record.

The **Review training data** section displays the detected entities, and the **NLP analysis** section displays the detected topic and entity types for the training data record.

6. To correct the training data for the IVA, perform one of the following actions:

Choices	Action
Create a new entity	a. In the Review training data section, in the data record content, highlight and right-click the text that you want to map to the new entity, and then click New entity. b. In the Create new entity window, in the Entity name field, enter a name for the entity. c. Click Submit.
Correct a detected entity	a. In the Review training data section, in the data record content, highlight and right-click the text for an existing entity. b. Select the name of another entity.

Choices	Action
Correct a detected topic	a. In the Topic field in the NLP analysis section, press the Down arrow key. b. Select a more appropriate topic for the training record.
Correct a detected language	In the Language list in the NLP analysis section, select a language for the training record.
Edit a data record	a. In the Review training data section, click the More icon, and then click Edit. b. In the Update text window, edit the text for the training record. For example, remove any trailing white spaces or correct spelling mistakes in the text. For more information, see Best practices for cleaning up training data in an IVA (on page 213). c. Click OK.

7. **Optional:** To use this training record to improve the artificial intelligence of the IVA, in the **Review training data** section, click **Mark reviewed**.
8. Correct additional records for the IVA by repeating steps [5 \(on page 208\)](#) through [7 \(on page 209\)](#).
9. Click **Save**.

[Training the model for the IVA channel \(on page 205\)](#)

Transferring training data to another IVA

When you generate training data for Pega Intelligent Virtual Assistant™ (IVA) that results in greater accuracy of topic and entity detection, you can reuse the information by transferring the training data to another IVA. Copying training data from one environment to another results in a quicker development of artificial intelligence of another chatbot making the system easier to build since you do not have to create the text analytics model from scratch. For example, you can copy accumulated training data from a production environment for a chatbot to a development or a QA environment of another chatbot. Before you can begin transferring data, you first start recording training data in the IVA.

Copying training data from one IVA environment to another involves first creating training data in a production environment, and then exporting the training data to a file. Next, you edit the IVA in a



development or QA environment, and then import the saved training data from the file. You can export either only reviewed training data for an IVA, or all of the training data. The training data that you export includes entities that the system added to the text analyzer, but did not build in the model.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
 2. In the navigation pane of App Studio, click **Channels**.
 3. Save training data from a chatbot:
 - a. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel, for example, in a production environment.
 - b. In the channel, click the **Training data** tab.
 - c. Click the **More** icon above the list of training records, and then click **Export data**.
-  **Note:** You can export data only if the IVA contains at least one training record.
- d. In the **Export data** window, select one of the following options:
 - To export only the reviewed data, click **Reviewed only**.
 - To export all data for the model, click **Reviewed and unreviewed**.
 - e. Click **Export**.
 - f. Click **Download** and save the training data as a **.zip** file.
4. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
 5. In the navigation pane of App Studio, click **Channels**.
 6. Reuse training data in another chatbot:
 - a. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel, for example, in a development or QA environment.
 - b. In the channel, click the **Training data** tab.
 - c. Click the **More** icon above the list of training records, and then click **Import data**.
 - d. In the **Import data** window, click **Choose file**, and then select the **.zip** file that contains the exported training data.
 - e. Click **Import**.

Training the model for the IVA channel (on page 205)

Applying changes to a text analytics model for an IVA

To update Pega Intelligent Virtual Assistant™ (IVA) with training records that you have reviewed, so that the system extracts the correct information from a chat conversation, apply the modified records to the text analytics model. By applying these changes, you rebuild the model with the additional training data, which improves the accuracy of IVA detection of topics and entities in user input.

The IVA records training data from a chat conversation when you enable the recording of training data, or when you create data records manually for the system.



Note: You can apply changes to the text analytics model only if training data exists for the IVA channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. In the channel, click the **Training data** tab.
5. In the list of training records, select the checkboxes for the records that you want to add to the model, and then click **Mark reviewed**.
6. Click the **More** icon next to the **Add records** button, and then click **Build model**.

[Training the model for the IVA channel \(on page 205\)](#)

[Text analysis concepts \(on page 21\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

Best practices when training model-based entities in an IVA

To train Pega Intelligent Virtual Assistant™ (IVA) to extract correct information from a conversation, you can modify the data records by highlighting relevant entities. For the IVA to respond more efficiently during a chat conversation, you train the analytics model to identify a word or phrase as structured content, and then associate metadata with that content. For example, you can train the system to detect the make and model of a car by identifying this information in the data records as entities.

When training model-based entities in Pega Platform, ensure that you highlight multiple types of entities in one phrase, sentence, or document. This is because the model learns the probability of an entity occurring, and the probability of the entity being followed or preceded by other entity types, based on the accompanying words. Additionally, the model uses the training data to learn the order in which entities are likely to appear. Apart from single words or short phrases, you can also mark larger chunks of text as



entities, for example, entire paragraphs. When you are training the model for the IVA, highlight at least 10 to 20 instances per entity type in the text. Try to avoid highlighting incomplete tags, missing characters, or misspelled words. Any mistakes affect the accuracy of the model.



Note: To learn more about best practices for creating entities for a text analytics model, see [Best practices for creating entity extraction models \(on page 205\)](#).

[Training the model for the IVA channel \(on page 205\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

Best practices when building rule-based entities in an IVA

To help Pega Intelligent Virtual Assistant™ (IVA) extract the correct information from a conversation, create rule-based entities by using Apache Ruta (Rule-based Text Annotation) scripts. Ruta is a rule-based script language with which you detect keywords and phrases that follow certain patterns. For example, when you build rule-based entities using Ruta scripts, the IVA detects information such as email addresses, so that the system can later populate this data in case properties.

The Ruta script lists entities based on rules that are a combination of annotation patterns, optional quantifiers, conditions for matching, and actions to perform. Ruta scripts are similar to regular expressions in that you define a pattern to compare against, so that you can obtain matching keywords and phrases. Examples of patterns that you can search for as rule-based entities include country names, postal codes, email addresses, phone numbers, and street names. To train an IVA to detect more abstract keywords and phrases in a conversation, train the data using model-based entities. For more information, see [Best practices when training model-based entities in an IVA \(on page 211\)](#).

When building a rule-based entity, remember that a Ruta script can detect only a single entity type. To store annotation results, mark them in the Ruta script. You can use variables, for example, *VarA* or *VarB*, to store intermediate annotation results. Keep in mind that you can also define wordlists as keywords and refer to them in the Ruta script. For more information, see [Best practices for pattern extraction in text analytics \(on page 205\)](#).

You define rule-based entity types with Ruta scripts that match a certain pattern, in Prediction Studio, during the creation of a new extraction model. To use this text analytics model in an IVA, you later select the model in the text analyzer rule for your channel. In addition, you can also create simple keyword-based entities when you create a new model. Use the keyword detection method when the entity type that you want to extract is an umbrella term for a finite number of associated terms or phrases that do not follow any specific pattern. For example, you can define and associate the *city* entity type with the keyword

Chicago, and the keyword with such synonyms as *Chi*, *The Loop*, and *The Windy City*. For more information about creating rule-based and keyword-based entities in Prediction Studio, see [Creating entity extraction models \(on page 20\)](#).

[Training the model for the IVA channel \(on page 205\)](#)

[Exploring text predictions and text analyzers \(on page 23\)](#)

[Text analysis concepts \(on page 21\)](#)

Best practices for cleaning up training data in an IVA

Before you apply changes to the text analytics model for Pega Intelligent Virtual Assistant™ (IVA), ensure that you correct each training data record by fixing issues in the text and removing any misplaced characters in the content. Correcting training data ensures that the IVA learns only from properly formatted sample data. Eliminating mistakes when you train the IVA helps to improve the accuracy of the model.

To clean up each training record for an IVA, ensure that you remove trailing white spaces, non-alphanumeric characters, and typos. You can also eliminate incomplete tags, missing characters, and misspelled words. For example, you can remove from a data record such characters as # or misplaced apostrophes and quotation marks in sentences and phrases. To learn more about editing a training data record, see [Correcting training data in an IVA \(on page 207\)](#).

[Training the model for the IVA channel \(on page 205\)](#)

Troubleshooting the conversational channel

Perform tasks in Pega Intelligent Virtual Assistant™ (IVA) to address problems, troubleshoot accessibility issues, adjust the accuracy of chatbot responses, and ensure that the system works as expected. As a result, you improve the quality and health of your chatbot, which helps users to interact with your application more easily when they want to resolve their problems. For example, you can help the system become more responsive to users by addressing an issue in which the chatbot is not always detecting the correct topics.

Channel developers and channel administrators update, troubleshoot, and maintain the IVA when the system is in a development, QA, or production environment.



Note: To learn more about troubleshooting natural language processing (NLP) issues in your chatbots, see also [Troubleshooting NLP](#).

- Fix general issues related to all of the conversational channels.
For more information, see [Troubleshooting conversational channel issues \(on page 214\)](#).
- Fix issues related to the IVA for Digital Messaging (Apple Business Chat, Facebook, SMS/MMS, and WhatsApp).
For more information, see [Troubleshooting the Digital Messaging channel \(on page 217\)](#).
- Fix issues related to the IVA for Legacy Webchat.
For more information, see [Troubleshooting the Legacy Webchat channel \(on page 219\)](#).
- Copy your legacy channel to the Digital Messaging channel to ensure that you reach users on additional social messaging platforms, for example, Apple Business Chat, or WhatsApp.
For more information, see [Updating an existing channel to the Digital Messaging channel \(on page 221\)](#).

[Pega Intelligent Virtual Assistant overview \(on page 14\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

[Conversational channels \(on page 4\)](#)

Troubleshooting conversational channel issues

Resolve general issues that occur in all Pega Intelligent Virtual Assistants (IVAs) so that the system works as expected and users can interact with chatbots. By troubleshooting your chatbot during the development, QA, and production stages, you ensure accurate interactions with users by means of text analysis and artificial intelligence. For example, for a bank loan application, if the chatbot is accessible at all times, the system can automatically start a business case for each request for a bank loan.

You might encounter the following issues:

- [The chatbot does not respond to user input \(on page 214\)](#)
- [Creating or processing a business case does not work \(on page 215\)](#)
- [The chatbot does not extract entities \(on page 215\)](#)
- [The chatbot generates an error when you use an existing paragraph rule as text for a response command \(on page 216\)](#)

A channel developer and the channel administrator troubleshoot issues reported with the chatbot.

The chatbot does not respond to user input

When a channel developer or a channel administrator starts a conversation in the preview console, or a chat window for an IVA in a staging environment, for example, in Facebook Messenger or WhatsApp, the system does not respond to the entered text.

- Ensure that you configure a unique operator with a valid access group in the Digital Messaging channel settings. This template operator must have access to the application in which you defined the Digital Messaging channel and use a valid access group.
For more information, see [Template operator for IVA channel \(on page 171\)](#) and [Learning about access groups \(on page 171\)](#).
- If the system does not display a paragraph for the response in the preview console, use the Live UI tool to find the referenced paragraph in the user interface, and then in the paragraph settings check whether the paragraph exists in the correct ruleset.
For more information, see [Live UI \(on page 171\)](#).
- If there is an issue with the natural language processing (NLP) of user input, temporarily disable or remove all the text analyzers from the advanced configuration, and then check that you obtain a response when the system uses only the default exact match text analyzer. When you do get a response, improve the system by adding training records and rebuilding the text analytics model for your advanced text analyzers. You can perform these tasks from Prediction Studio or from the conversational channel.
For more information, see [Adding a text analyzer for an IVA \(on page 188\)](#).
- Remember that each conversational channel, such as Apple Messages for Business, Facebook, Legacy Webchat, and WhatsApp, has different capabilities. For your reference, always refer to the specific messaging channel documentation when creating and building a chatbot.

Creating or processing a business case does not work

When a channel developer and channel administrator interacts with a chatbot, the system cannot process or create a new business case that contains information from the conversation.

- Follow these guidelines to check that you did not create an invalid case definition:
 - The business case cannot contain user interface forms. Build the case using the conversational builder, using the Question smart shape that is based on flows.
 - The business case must reach the *Resolved* status or be reassigned to another operator.
 - The business case must generate at least one question. If you have not defined at least one question for the case, the case will terminate with an error response.

For more information, see [Adding a case type conversation process for a conversational channel \(on page 179\)](#).

- Ensure that you did not create the business case outside of the current interaction. The system only processes business cases that are created in the same interaction.

The chatbot does not extract entities

While testing your chatbot in the preview console, the system does not detect and extract the defined entities during a chat conversation. A channel developer uses the preview console to verify whether the

system correctly detects the entities. For example, the chatbot only extracts entities from text that is entered in the chat if a business case was started in the system.

Remember that the result of entity extraction is always the creation of a work item, in other words, a business case in the application. The chatbot then asks a series of questions and detects entities in the responses.

- Depending on your use case, ensure that in the configuration of your advanced text analyzers, you specify that the system detects entities outside of a case, in the context of a case, or both.

For more information, see [Adding a text analyzer for an IVA \(on page 188\)](#).

- Improve the text analysis for the system by adding training records with sample entities to detect, and then rebuilding the text analytics model.

For more information, see [Training the model for the IVA channel \(on page 205\)](#) and [Correcting training data in an IVA \(on page 207\)](#).

The chatbot generates an error when you use an existing paragraph rule as text for a response command

When you edit a paragraph rule that you will use in the chatbot as a template for the response command, you might click the **Insert Rule** icon in the toolbar, and then add another rule as a section that contains a complex form, advanced HTML, and embedded JavaScript actions in the paragraph. After you save the paragraph rule and later attempt to use this existing template containing a complex form with embedded sections, the chatbot generates an error.

- Follow these guidelines when editing a paragraph rule to use as text for a chatbot response command:
 - Ensure that the paragraph rule only contains information that you define using simple HTML, for example, for a standard form.
 - You can add images and links to the text in the paragraph rule.
 - Do not use the **Insert Rule** icon on the toolbar to modify the contents of the existing paragraph rule, for example, by embedding another complex paragraph rule as a section.
 - Do not use JavaScript code in the paragraph rule.
 - Do not select a section or custom HTML as the question layout when editing a question rule that you will use for a chatbot response command. If you select rich text or a paragraph as the question layout, ensure that the rule only contains information defined using simple HTML, without embedded rules and sections.
- For security reasons, starting from Pega Platform version 8.3.2, adding complex information in a paragraph rule for a chatbot response command is not supported, for example, JavaScript actions and advanced HTML in embedded forms.

[Troubleshooting the Digital Messaging channel \(on page 217\)](#)

[Troubleshooting the Legacy Webchat channel \(on page 219\)](#)

[Troubleshooting the conversational channel \(on page 213\)](#)

Troubleshooting the Digital Messaging channel

Fix issues with Pega Intelligent Virtual Assistant™ (IVA) for Digital Messaging (Apple Messages for Business, Facebook, SMS/MMS, and WhatsApp) to make the system available to users at all times so that they can ask questions and report issues in the application. For example, you can resolve secure communication problems for the channel, improving the general accessibility of the chatbot.

A channel developer and the channel administrator troubleshoot reported issues in the chatbot. You might encounter the following issues:

- The system does not show new Digital Messaging Manager features after you update Pega Platform or Pega Customer Service. [\(on page 217\)](#)
- The system displays a warning message stating that you need to define Digital Messaging security settings [\(on page 218\)](#)
- The Digital Messaging channel definitions disappear from the application [\(on page 218\)](#)
- The system displays an error when you try to add messaging channels to another Digital Messaging channel [\(on page 219\)](#)
- The Digital Messaging service cannot communicate with Pega Platform [\(on page 219\)](#)

The system does not show new Digital Messaging Manager features after you update Pega Platform or Pega Customer Service.

To use the latest Digital Messaging Manager features, save your existing Digital Messaging channel after a Pega Platform™ or Pega Customer Service™ update.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging channel.
4. Click **Save**.

The system displays a warning message stating that you need to define Digital Messaging security settings

During the creation and configuration of a Digital Messaging channel in the Pega Platform application, the system displays the following warning message:

Define Digital Messaging security within the application definition. Couldn't retrieve the connection details.

Contact Pega Support to obtain the security settings for your Digital Messaging channel, including the **Manager ID** and the **Manager Key** values for the Digital Messaging Manager. You specify these settings in the application rule on the **Security** tab.

For more information, go to [Pega Support](#) and also see [Configuring Digital Messaging channel security \(on page 168\)](#).

The Digital Messaging channel definitions disappear from the application

The system removes the configured social messaging channels from Digital Messaging Manager for the Digital Messaging channel.

This issue occurs when you either update the **Public Link URL** field of your Pega Platform or change the values for the **Manager ID** or **Manager Key** in the application rule, after you create a channel and configure its IVAs for Apple Messages for Business, Facebook, SMS/MMS, or WhatsApp in the Digital Messaging Manager. When you update the **Public Link URL** field or the Digital Messaging security settings for your channel, the system automatically generates a new unique consumer ID. As a result, all of your previously configured social messaging channels in the Digital Messaging Manager are no longer available.

- If you need to change the **Manager ID** or **Manager Key**, it is important to first deregister your messaging channels. Before updating your Digital Messaging channel configuration in the application rule on the **Security** tab, ensure that you first remove your messaging channels for the consumer ID in Digital Messaging Manager for each IVA, for example, for Facebook, or WhatsApp. After you remove all of your messaging channels, you can safely update the **Manager ID** or **Manager Key**.

For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).

- If you have updated the **Manager ID** or **Manager Key** before you removed your messaging channels, contact Pega Support and request that they remove all of your existing channels. In your service request, specify that your **Manager ID** or **Manager Key** was changed and you can no longer access your messaging channels.

For more information, go to [Pega Support](#).

- Ensure that you did not modify the **Public Link URL** field that is referencing your Pega Platform instance.

For more information, see [Public URL for Directed Web Access \(on page 214\)](#).

The system displays an error when you try to add messaging channels to another Digital Messaging channel

When you attempt to add messaging channels that are already defined in an existing Digital Messaging channel to another channel, the operation fails and the system displays an error message.

1. Ensure that you first deregister the messaging channels, for example, Facebook, or WhatsApp from Digital Messaging Manager for the existing Digital Messaging channel.

Note: Deleting a Digital Messaging channel from your system automatically deregisters all of its defined messaging channels from Digital Messaging Manager, for example, Facebook, or WhatsApp. For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).

2. Add the messaging channels to another Digital Messaging channel.

For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).

The Digital Messaging service cannot communicate with Pega Platform

If your Pega Platform system was installed on premises, ensure that you specified a value in the **Public Link URL** field of your system settings so that the Digital Messaging service can communicate with Pega Platform.

Define a value in the **Public Link URL** field for your Pega Platform instance.

For more information, see [Public URL for Directed Web Access \(on page 214\)](#).

[Troubleshooting conversational channel issues \(on page 214\)](#)

[Troubleshooting the conversational channel \(on page 213\)](#)

Troubleshooting the Legacy Webchat channel

Ensure that Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat works as expected and is accessible to users so that they can interact with your application. By making your chatbot available at all times, you ensure that your customers have access to key information and can resolve their issues more quickly.



Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

You might encounter the following issues:

- [I do not get any response from the chatbot \(on page 220\)](#)
- [I do not know how to style the IVA for Legacy Webchat \(on page 220\)](#)

A channel developer and the channel administrator troubleshoot issues reported with the chatbot.

I do not get any response from the chatbot

When I enter text in the chat window, I do not get any response from the IVA for Legacy Webchat.

- Do not use the same web browser for your instance of the Pega Platform application that you use for the chat window for the IVA for Legacy Webchat. The chatbot uses cookies to handle authentication, so the system will log you in as the same user as for your application, causing this issue. Ensure that you open your Pega Platform instance in another web browser.
- Check the mashup security settings for your application, by performing the following steps:
 1. Switch to Dev Studio.
 2. In the header of Dev Studio, click the name of the application, and then click **Definition**.
 3. In the application rule, click the **Integration & security** tab.
 4. In the **Mashup security** section, verify that there are no limitations on the mashup used by the chat window for your IVA.
 5. Click **Save**.
 6. Switch back to App Studio.

I do not know how to style the IVA for Legacy Webchat

I want to modify the user interface for the IVA that the chat window and the preview console use, to fit my organization's specific needs and requirements.

1. Change the CSS styles for the chat window for the IVA for Legacy Webchat by editing the *pyWebChatCustomization* rule.

For more information, see [Customizing the Legacy Webchat channel \(on page 224\)](#).



2. Modify the default text for the action prompt that is displayed in the preview console and the chat window for the IVA.
For more information, see [Changing the default text for action prompt in the IVA for Legacy Webchat \(on page 226\)](#).
3. Edit the Legacy Webchat channel sections that user input and responses use, and make user interface changes in the sections.
For more information, see [Customizing the Legacy Webchat channel \(on page 224\)](#).

[Troubleshooting conversational channel issues \(on page 214\)](#)

[Troubleshooting the conversational channel \(on page 213\)](#)

Updating an existing channel to the Digital Messaging channel

Copy your existing legacy Pega Intelligent Virtual Assistant™ (IVA) to a Digital Messaging channel so that users can interact with your application on additional social messaging platforms. The IVA for Digital Messaging provides a better overall user experience and adds versatility to your system because you can build chatbots for Apple Messages for Business, Facebook, SMS/MMS, or WhatsApp from a single channel. For example, if you already created an IVA for Legacy Webchat and successfully built its text analytics model, you can copy the legacy channel to a new Digital Messaging channel, to ensure that the chatbot for your application also interacts with users on Apple Messages for Business, and WhatsApp.

To successfully copy your Legacy Webchat channel to a Digital Messaging channel, open the legacy channel and select a command to copy the channel to a new Digital Messaging channel, then enable Digital Messaging-specific conversations for your case types, and then define the social messaging platforms that you want to use in the new channel.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Legacy Webchat channel.
4. Click **Actions**, and then click **Copy channel interface**.
5. In the **Copy channel** window, click **Digital Messaging**.

The system displays a Digital Messaging channel.

6. Configure the new Digital Messaging channel and save your changes.
Do not set up the channel for any social messaging platforms yet. For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).

The system creates a new Digital Messaging and copies the text analyzer and its customizations, case commands, response commands, and other settings from the Legacy Webchat channel.

7. Enable Digital Messaging-specific conversations for your case types that are enabled for your existing legacy channel.

For more information, see [Adding a conversational channel to a case type process \(on page 180\)](#).

8. Define social messaging platforms for your new Digital Messaging channel.

For more information, see [Creating a Digital Messaging channel \(on page 113\)](#).

[Troubleshooting conversational channel issues \(on page 214\)](#)

[Troubleshooting the conversational channel \(on page 213\)](#)

Creating a Legacy Webchat channel

Create Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat so that users can interact with a Pega Platform™ application to answer questions, report an issue, or step through a process by sending messages in an embedded chat window. For example, with the IVA for Legacy Webchat, users can create a case to inquire about a car price estimate or book a flight.

Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

An IVA for Legacy Webchat takes advantage of similar bot capabilities that are available in the IVA for Facebook. However, you have full control over the end-to-end communication and no additional accounts or third-party installation is necessary.

Note:



- A chatbot works in one language because the conversational channel uses a template operator that is defined for a single locale. For more information, see [Template operator for IVA channel \(on page 171\)](#).
- If you update your chatbot to the 8.7 version and default language in the previous version was not English, for example, German, ensure that on the updated 8.7 version of your system you make German the first language in the list of supported languages. By default, in an updated chatbot to the 8.7 version, the default first language in the list is always English.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Create new channel interface** section, click **Legacy Webchat**.
4. In the **Details** section, in the **Channel interface name** field, enter a name for your channel.
5. **Optional:** To describe the purpose of the channel, in the **Description** field, enter a short description of your IVA.
6. In the **Supported languages** list, select the languages in which the chatbot interacts with users. You select the languages in which the chatbot processes user interactions so that the system can perform text analysis and provide automatic responses in the correct language. The first language in the list is the default language and acts as the fallback language for the chatbot in a case no translation is yet available in another language.
7. In the channel, click the **Connection** tab.
8. In the **Template operator ID** list, click the name of an existing Pega Platform operator to use as a template.
The system uses this operator as a template to create all further accounts. Each account has a unique channel identifier set as its ID.
9. Click **Save**.
10. Embed a Legacy Webchat channel chat window on a web page.
For more information, see [Embedding a chat window for the Legacy Webchat channel \(on page 223\)](#).

[Template operator for IVA channel \(on page 171\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

Embedding a chat window for the Legacy Webchat channel

Generate mashup code for your Legacy Webchat channel and embed it on an external website like a webpage. The webpage displays a chat window where users can interact with a Pega Platform application, for example, to obtain a list of commands or open a case.



Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

To generate the mashup code for a website, specify a URL address for your Pega Platform application so that the system knows where to connect during a chat conversation.

Note: Generate the mashup code again and embed it on a website, if you change any Legacy Webchat channel settings, for example, the URL for your Pega Platform application.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Legacy Webchat channel.
4. In the Legacy Webchat channel interface, click the **Mashup** tab.
5. In the **Mashup URL** field, enter the URL for your Pega Platform application.
6. Click **Save**, and then click **Generate**.
7. Copy and save the generated code from the **Mashup code** text area, for example, click **Copy to clipboard**.
8. Embed the mashup code from step 7 ([on page 224](#)) on a webpage that will be used as an IVA for Legacy Webchat.

[Creating a Legacy Webchat channel \(on page 222\)](#)

Customizing the Legacy Webchat channel

Modify the styling of Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat to fit your business needs and requirements. You can customize the Cascading Style Sheets (CSS) styles, change the text for the action prompt for the chatbot, customize a user interface section used for chat responses, and also add custom controls to the channel.

Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

For example, add your own icons or additional useful information, such as date and time stamp for each chat message.

Note: The styling changes that you make in the system affect all Legacy Webchat channels that are defined in a Pega Platform application.

1. Switch to Dev Studio.
2. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Legacy Webchat channel.
4. Modify the default text for the action prompt that is displayed in the preview console and chat window for the IVA.
For more information, see [Changing the default text for action prompt in the IVA for Legacy Webchat \(on page 226\)](#).
5. Update the Legacy Webchat channel section used for user input.
 - a. Edit the pyRenderSimulatorInbound section.
 - b. Make user interface changes in the section.
 - c. Click **Save**.
6. Update the Legacy Webchat channel section used for the response.
 - a. Edit the pyRenderSimulatorOutbound section.
 - b. Make user interface changes in the section.
 - c. Click **Save**.
7. Change the CSS styles for the Legacy Webchat channel.
 - a. Edit the pyWebChatCustomization rule.
 - b. Modify the CSS styles, for example, for the background color, client input, submit area, header text, or submit button.
 - c. Click **Save**.
8. Create custom controls in the Legacy Webchat channel.

For more information, see [Creating custom controls for the IVA for Legacy Webchat \(on page 226\)](#).

9. Switch back to App Studio.

[Creating a Legacy Webchat channel \(on page 222\)](#)

Changing the default text for action prompt in the IVA for Legacy Webchat

To make Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat more user friendly, you can modify the default text for the action prompt that the system displays to users in the chatbot window and the preview console. As a result, you also help to customize the look and feel of the chatbot to better fit the needs and requirements of your organization.

Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

1. Switch to Dev Studio.
2. In the navigation pane of Dev Studio, click **App**.
3. Click **Classes**.
4. In the field below the **Classes** tab, search for the *Embed-Channel-Control-* class.
5. In the *Embed-Channel-Control-* class, expand the **Data Model > Field Value** section, and then double-click the *pyActionPrompt pyInputPlaceholder* rule.
6. In the opened rule, in the **To** field, enter new text for the default prompt for the chatbot.
By default, the action prompt in the chatbot displays the following text: Type your message here
7. Click **Save**.
8. Switch back to App Studio.

[Creating a Legacy Webchat channel \(on page 222\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

Creating custom controls for the IVA for Legacy Webchat

Improve how users interact with Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat by implementing your own user input controls for the chatbot. By implementing custom controls you ensure a better overall user experience, and that the IVA meets the needs and requirements of your organization.



For example, when a chatbot displays your custom controls, such as an autocomplete field, users can obtain help and resolve their issues more quickly.

Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

To create your own custom control for the IVA for Legacy Webchat, define a class that inherits from *Embed-Channel-Control-* in the following format: *Embed-Channel-Control-<custom_control_name>*. For example, to create a custom control that is used as an autocomplete field, you create the *Embed-Channel-Control-Autocomplete* class in the system. For more information, see [Working with custom controls in the IVA for Legacy Webchat \(on page 230\)](#).

1. Switch to Dev Studio.
2. Create a new abstract class for your custom control in the following format: *Embed-Channel-Control-<custom_control_name>*.
3. Open the *Embed-Channel-Control-.pyDefault* rule, save this data transform rule as *Embed-Channel-Control-<custom_control_name>.pyDefault*, set *pxIsSimpleMode* to *false*, and clear the **Call superclass data transform** checkbox.
4. Override the *Work-Channel-Interaction.pySetInputControl* rule and trigger the *pxCreateControlOfClass* activity with *Embed-Channel-Control-<custom_control_name>* as its parameter.

Note: Ensure that the chatbot does not always use this custom control by configuring the *Embed-Channel-Control-<custom_control_name>* rule to run only under certain conditions.

5. **Optional:** To use a different type of input in the chatbot for the custom control, perform the following steps:
 - a. Open the *Embed-Channel-Control-.pyInput* section rule.
 - b. Save this rule as the *Embed-Channel-Control-<custom_control_name>.pyInput* section.
 - c. Edit the section rule to customize the input, and then save your changes.

Note: By default, the *.pyText* property is the value that the system sends to the chatbot. If you have not yet customized the **Send** button action, the control is based on this value.



However, if you customize this section, ensure that the chatbot correctly handles events for the **Send** button. For more information, see [Configuring a Text input control \(on page 228\)](#).

6. **Optional:** To customize the style and behavior of the **Send** button in the custom control, perform the following steps:
 - a. Open the *Embed-Channel-Control-.pySubmitButton* rule.
 - b. Save this rule as the *Embed-Channel-Control-<custom_control_name>.pySubmitButton* section.
 - c. Edit the section rule to customize the **Send** button and save your changes.



Note: If you want the chatbot to process the message, trigger the *pxProcessSimulation* activity. To trigger this activity, ensure that you check the default action that is configured on the default button implementation.

7. **Optional:** To add more action buttons in the custom control, perform the following steps:
 - a. Open the *Embed-Channel-Control-.pyActionButtons* rule.
 - b. Save this rule as the *Embed-Channel-Control-<user_control_name>.pyActionButtons* section.
 - c. Edit the section rule to add more action buttons and save your changes.



Note: By default, the IVA includes the **Add attachment** icon. You can add more action buttons in the section that the system displays below the input area of the chat window.

8. **Optional:** To switch between the simple and advanced mode shown in the default text area of the custom control, perform the following steps:
 - a. Open the *Embed-Channel-Control-.pyOnSwitchToSimpleMode* activity.
 - b. Save this rule as the *Embed-Channel-Control-<custom_control_name>.pyOnSwitchToSimpleMode* activity.
 - c. Edit the activity rule to switch to simple mode and save your changes.
These steps ensure that you provide a value conversion when the *pxSwitchToSimpleMode* API is called, no matter whether the system uses a standard control or your custom control.
 - d. Open the *Embed-Channel-Control-.pyOnSwitchToAdvancedMode* activity.
 - e. Save this rule as the *Embed-Channel-Control-<custom_control_name>.pyOnSwitchToAdvancedMode* activity.

- f. Edit the activity rule to switch to advanced mode and save your changes.

These steps ensure that you provide a value conversion when the *pxSwitchToSimpleMode* API is called, no matter whether the system uses a standard control or your custom control.



Note: You can implement buttons to switch between the simple and advance modes in your chatbot by using the *pxSwitchToSimpleMode* API activity to switch the view to the default text area. You can also use the *pxSwitchToAdvancedMode* API activity to switch the view to *Embed-Channel-Control-<custom_control_name>.pyInput*. For more information about configuration for controls such as the date control, see *Configuring common control settings (on page)*.

9. **Optional:** To provide additional conversion of text entered in the custom control before the system submits this user input to the chatbot for processing, perform the following steps:
 - a. Open the *Embed-Channel-Control-.pyOnSubmit* activity.
 - b. Save this activity rule as *Embed-Channel-Control-<custom_control_name>.pyOnSubmit*.
 - c. Edit the activity rule to provide additional conversion for the entered text, and then save your changes.



Note: Performing additional conversion of user input is useful when there is no need to update the **Send** button, as the system calls this extension by default for all of the standard built-in controls used by the chatbot.

10. **Optional:** To enable the addition of attachments for the custom control, perform the following steps:
 - a. Open the *Embed-Channel-Control-.pyAttachmentAddingEnabled* when rule.
 - b. Save this rule as *Embed-Channel-Control-<custom_control_name>.pyAttachmentAddingEnabled*.
 - c. Modify the when rule so that it returns `true`, and then save your changes.
11. **Optional:** To customize a section that the system displays above the input area in the chat window, perform the following steps:
 - a. Open the *Embed-Channel-Control-.pyActionsBar* section rule.
 - b. Save this rule as the *Embed-Channel-Control-<custom_control_name>.pyActionsBar* section.
 - c. Edit the section rule used for the input area, and then save your changes.
 - d. To make the custom section visible when your custom control is used, open the *Embed-Channel-Control-.pyShowActionsBar* when rule.

- e. Save this rule as *Embed-Channel-Control-<custom_control_name>.pyShowActionsBar*.
- f. Modify the when rule so that it returns `true`, and then save your changes.



Note: The chatbot displays a custom section above the input area that contains standard controls, such as quick replies.

12. Switch back to App Studio.

[Customizing the Legacy Webchat channel \(on page 224\)](#)

[Creating a Legacy Webchat channel \(on page 222\)](#)

Working with custom controls in the IVA for Legacy Webchat

Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat is based on the Pega Platform user interface. You embed the chatbot user interface as a standard chat window by using a web mashup on your website. As a result, you can use standard Pega user interface controls or optionally create your own controls to fit the needs and requirements of your organization. By creating custom controls for your chatbot, you improve the overall user experience of the application and help users address their issues quicker and more efficiently.



Note: Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

The IVA includes a mechanism to help channel developers implement custom input controls that the system displays to users while they interact with the chatbot. You can also override the default mechanism used to display the standard built-in controls, such as a date or integer. By default, the chatbot uses a mechanism that is based on property types. For example, if the property of a question asked by the chatbot is a date type, then the system displays the date and time control so that users can pick a date using the Date picker control.

Chatbot user interface controls

Embed-Channel-Control- is the base abstraction class for all standard built-in controls that the IVA for Legacy Webchat uses:



- *Embed-Channel-Control-Date* - Defines the date control so that users can enter a date in the chatbot.
- *Embed-Channel-Control-DateTime* - Defines the date and time control so that users can enter a date and time in the chatbot.
- *Embed-Channel-Control-Decimal* - Defines the decimal control so that users can enter a decimal number in the chatbot.
- *Embed-Channel-Control-Form* - Defines the form control so that users can fill out a simple form in the chatbot.
- *Embed-Channel-Control-Integer* - Defines the integer control so that users can enter an integer number in the chatbot.
- *Embed-Channel-Control-QuickReplies* - Defines the quick replies control so that users can quickly select one of several options displayed in the chatbot.
- *Embed-Channel-Control-Text* - Defines the text control so that users can enter text in the chatbot.
- *Embed-Channel-Control-TimeOfDay* - Defines the time of the day control so that users can enter a time in the chatbot.
- *Embed-Channel-Control-TrueFalse* - Defines the true and false control so that users can enter a boolean value in the chatbot.

To create your own custom control for the IVA for Legacy Webchat, define a new class that inherits from *Embed-Channel-Control-* in the following format: *Embed-Channel-Control-<custom_control_name>*. For example, to create a custom control for an autocomplete field, you can create the *Embed-Channel-Control-Autocomplete* class in the system. You can also extend the standard controls listed above that are used by the chatbot. For example, to create your own date control that displays information in a format for the Aztec calendar, you can create a new class called *Embed-Channel-Control-Date-Aztec*. For more information, see [Creating custom controls for the IVA for Legacy Webchat \(on page 226\)](#).

Custom controls are one way to display user input in the chatbot. You can also customize the action set that is triggered for the **Send** button. In addition, you can provide a custom transformation in the system, from a custom control to a text control. However, these additional features are optional and depend on the requirements of your application. In most cases, you only implement your custom control in the chatbot so that it displays as a different type of input in the system.

Conditions under which the system displays the custom control

The system includes a mechanism that is an extension point, that you can use to determine which control to use based on the current state, for example, the current question in the chat window, or another condition.

Work-Channel-Interaction.pySetInputControl is an extension point with input parameters. By default, this extension point calls the *pxCreateControlOfClass* API which takes your custom control class as its

parameter. The *pySetInputControl* activity contains the *controlClass* input parameter that defines the built-in suggestion for which control to use, based on the property type. You can decide whether to pass this parameter to the *pxCreateControlOfClass* class or not. The class that you specify as the parameter in the *pxCreateControlOfClass* determines which control the system displays in the preview console and the chat window.

[Creating custom controls for the IVA for Legacy Webchat \(on page 226\)](#)

[Customizing the Legacy Webchat channel \(on page 224\)](#)

[Creating a Legacy Webchat channel \(on page 222\)](#)

Updating the access group for an IVA channel

You cannot authenticate from your Pega Intelligent Virtual Assistant™ (IVA) unless you set the access group for the *channelLogin* and *channelMgmt* service package rule to the access group for your Pega Platform application.



Note: By default, the access group for the *channelLogin* and *channelMgmt* service package rule is set to `PRPC:Administrators`.

1. Switch to Dev Studio.
2. In the navigation panel of Dev Studio, click **Records** to display the Records Explorer.
3. Click **Integration-Resources**, and then click **Service Package**.

The system displays a list of service package rules.

4. In the **Service Package Name** column, click the **channelLogin** rule.
5. In the service package rule, in the **Service access group** field, enter the access group for your Pega Platform application.
6. Click **Save**.
7. In the service package list, repeat steps 4 (on page 232) through 6 (on page 232) by editing the *channelMgmt* service package rule.
8. Switch back to App Studio.

[Creating an Alexa channel \(on page 261\)](#)

[Creating a Legacy Webchat channel \(on page 222\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)



Use case: Building an IVA for Legacy Webchat in the preview console

For quicker development of Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat, build the chatbot by simulating conversations in the preview console. Analyze the responses to your input to make instant changes in the chatbot's configuration, validate the updates, and build the text analytics model, so that the system is more responsive in the future.

In this tutorial, you will learn how to train and build the model to enhance the text analyzer engine that the chatbot uses, and to improve how the model responds based on artificial intelligence algorithms.

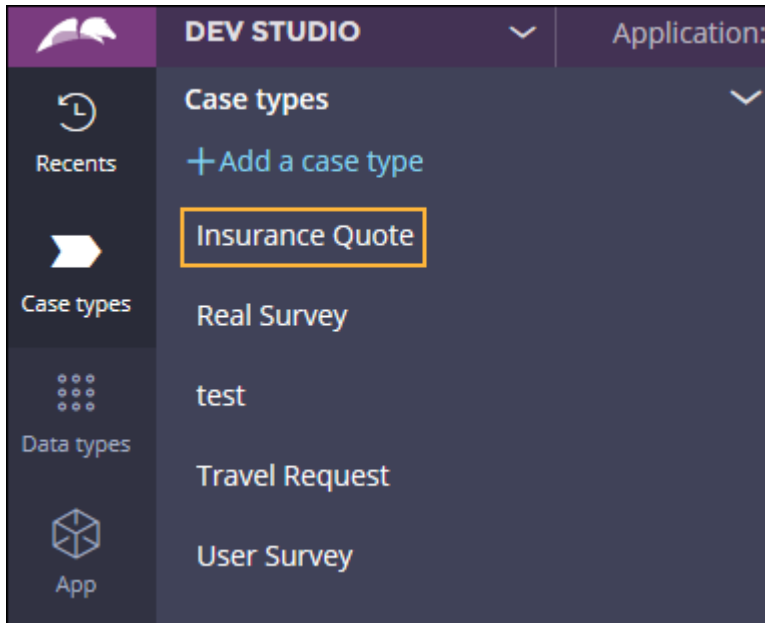
Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

Before you begin

To follow the steps in this tutorial, set up a sample Pega Platform application with the Insurance Quote case type and a Legacy Webchat channel.

Create the Insurance Quote case type for car insurance quotes with the following settings:

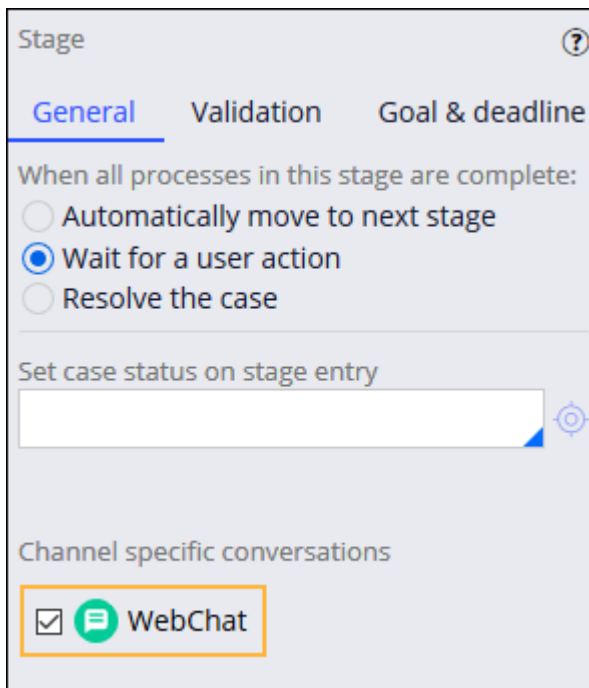
1. Add the following data types to the Insurance Quote case type for the car make, model, and vehicle identification number (VIN): *carMake*, *carModel*, and *carVIN*.



Case type explorer

For more information, see [Associating data objects with case types](#) (on page 234).

2. Enable Legacy Webchat channel-specific conversations for a stage of the Insurance Quote case type.



Stage configuration for the Insurance Quote case type

For more information, see [Adding a conversational channel to a case type process \(on page 180\)](#).

3. Configure conversation questions for a stage process of the Insurance Quote case type by adding three simple questions for the make, model, and VIN data types.

Conversation for WebChat-init

What is the make of your car?	Text (single line)
What is the model of your car?	Text (single line)
Please provide VIN number.	Text (single line)

[+ Add question](#)

Sample questions for the Insurance Quote case type

Configure a Legacy Webchat channel with the following settings:

- In the **Content** section on the **Configuration** tab, add the *insurance* create case command for the Insurance Quote case type.

Content

Create case command	Case type	Authentication
insurance	Insurance Quote	▶ ⚙️ 🗑️
travel	Travel Request	▶ ⚙️ 🗑️
survey	Real Survey	▶ ⚙️ 🗑️
+ Add case type		

Response command	Response type	Authentication
hi	Text	Required ▶ ⚙️ 🗑️
cancel	Cancel	▶ ⚙️ 🗑️
+ Add response		

Sample create case command for a case type

For more information, see [Adding case commands for a conversational channel \(on page 181\)](#).

- On the **Behavior** tab, enable advanced text analyzer configuration and add the iNLP text analyzer. Ensure that the IVA can interpret text in both the context of the case and outside of the case context.

Text Analyzer ☒ Use advanced configuration
Text analyzers that process user's input and look for matching responses.

Type	Apply in context of a case	Apply outside of case context	
1 Exact match	✓	✓	 
2 iNLP Edit topics	✓	✓	 

+ Add text analyzer

Enabled text analyzers

For more information, see [Adding a text analyzer for an IVA \(on page 188\)](#).

After you configure these settings for your IVA for Legacy Webchat, the system text analyzer engine can detect when a user requests a car insurance quote in the chat window. As a result, the chatbot will create the Insurance Quote case in the system. The system then asks the three insurance questions about the make, model, and VIN of the car.

Associating topics with user input

Improve the chatbot so that the system can correctly detect topics from user input by simulating a conversation in the preview console. You can associate existing topics with the user input that the chatbot does not recognize correctly, and add more examples of similar user input to improve future responses.

Topics are the general subject matter of a chat conversation that the system detects, for example, a request for a car insurance quote or an enquiry about a bank loan. After updating the text analytics model with samples of user input for the insurance topic, the chatbot correctly detects the subject matter and automatically starts the Insurance Quote case when the user requests information about a car insurance quote.

Associate topics with user input by performing the following steps:

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Legacy Webchat channel.
4. In the channel, click the **Behavior** tab.
5. In the preview console in the right panel, enter a request for a car insurance quote.

Preview console

☐ Show analysis Reset

Give me a car quote please Define topic

 Sorry, I did not understand that

Type something to try...

Sample car insurance question that is unassociated with any topics

The IVA for Legacy Webchat does not recognize the request for a car insurance quote because no topic is associated with the entered text.

6. Click **Define topic** below the entered text.
7. In the **Topic** field, press the Down arrow key, and then select the correct topic.
8. **Optional:** To associate the topic with additional similar user input, perform the following steps:
 - a. In the field below the topic name, enter more sample user input for the insurance topic.
 - b. Click **Add example** and add additional sample user input for the insurance topic.

Preview console

☐ Show analysis Reset

Give me a car quote please

Topic

insurance

I need car quote

how can I get car quote?

+ Add example

Cancel Submit

 Sorry, I did not understand that

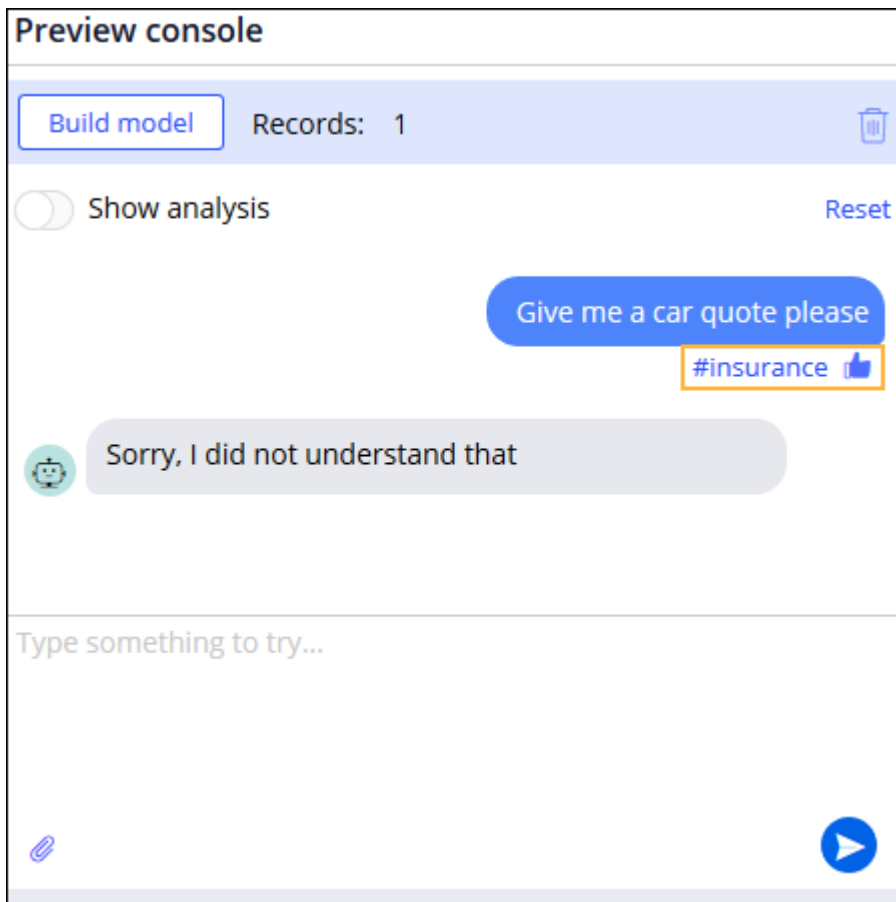
Type something to try...

Associating user input with a topic

c. To define additional user input, repeat step 8.b (on page 237).

9. Click **Submit**.



Sample topic that is associated with user input

The insurance topic is now associated with the user input that appears above the topic name.

10. Rebuild the model by clicking **Build model**.

The chatbot learns how to respond to this user input and the additional two examples that you provided. To further enhance the text analytics model, ensure that the correct topic is associated with the user input samples, and to improve the chatbot responses, add more examples of the same user input and rebuild the model a few more times.

Configuring and mapping entities to case type properties

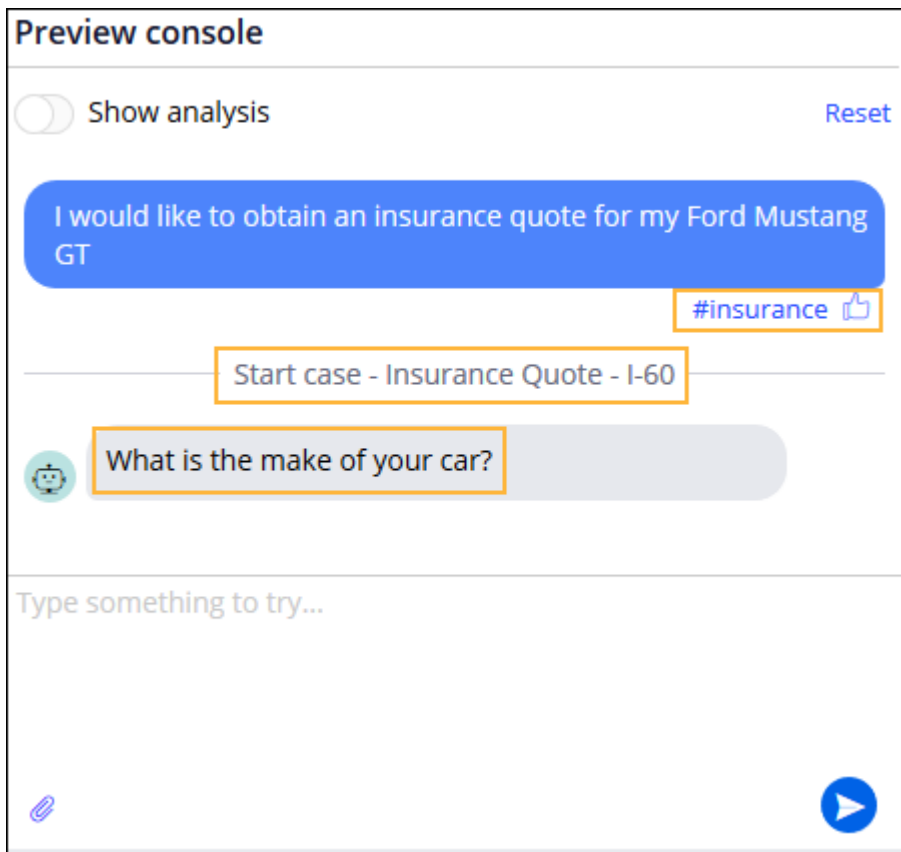
Train the chatbot to detect the correct entities from user input when a user requests an insurance quote, by simulating a conversation in the preview console. When you enter text and receive replies in the preview console, you can also define entities that are missing from user input. As a result, the system automatically adds the detected information about car makes and models from the actual chat conversations with users to the created Insurance Quote case properties.

Entities are short phrases detected in the conversation, for example, an email address, car make, or car model. In the example below, create the *Carmakeent* and *Carmodelent* entities to associate them with

the car make and model in the user input. Map these two entities to the *.carMake* and *.carModel* case properties that you define for the Insurance Quote case, respectively.

1. In the preview console, enter a request for an insurance quote.

Tip: If the correct topic is already associated with your input, you can improve the chatbot's future responses in the model by clicking the **Like** icon next to the topic name, and then rebuilding the model.



Sample request for a car insurance quote showing the detected topic and Like icon

Based on the insurance keyword, the chatbot correctly associates the text with the Insurance Quote case type, starts a case, and responds by asking the first question about the make of the car. The *#insurance* label below your request represents the topic, while the horizontal line indicates the start of case number I-60.

2. Display more information about the conversation and the case by turning on the **Show analysis** switch

Preview console

☒ Show analysis Reset

I would like to obtain an insurance quote for my Ford Mustang GT

#insurance

Create case: Insurance Quote

0.203 sec Neutral (0.0)

I would like to obtain an insurance quote for my Ford Mustang GT

Topic	Confidence
#insurance	97%

Start case - Insurance Quote - I-60

What is the make of your car?

Ask: carMake

Type something to try...

Sample user input analysis

The analysis shows that the system classified the sentiment of the text that you entered in the previous step as neutral, and that the insurance topic was guessed with 97 percent confidence. The chatbot did not detect any entities in your input, even though the text included the make of a car (Ford), and the model of a car (Mustang GT).

3. Associate new entities with "Ford" and "Mustang GT" from the text that you entered in step 1 (on page 240) by performing the following steps:
 - a. Select and right-click **Ford** in the displayed text, and then click **New entity**.
 - b. In the **Entity name** field, enter: Carmakeent
 - c. Click **Submit**.

- d. Select and right-click **Mustang GT** in the displayed text, and then click **New entity**.
- e. In the **Entity name** field, enter: Carmodelent
- f. Click **Submit**.

Preview console

Build model

Records: 1

☒ Show analysis

Reset

I would like to obtain an insurance quote for my Ford Mustang GT

#insurance

Create case: Insurance Quote

0.203 sec Neutral (0.0)

I would like to obtain an insurance quote for my

Ford Mustang GT

Entity types

Carmakeent Carmodelent

Topic	Confidence
#insurance	97%

Start case - Insurance Quote - I-60

What is the make of your car?

Ask: carMake

Type something to try...

Sample user input analysis with mapped entities

The system maps the two entity types that you create to the make and model of the car in the text, Ford and Mustang GT, respectively.

4. Map the extraction of the two entity types that you created to case properties, by performing the following steps:
 - a. In the **Create case** section, click the **Open response configuration** icon.
 - b. In the **Response configuration** dialog for the Insurance Quote case type, click the **Entities extraction** tab.
 - c. Click **Add mapping**.
 - d. In the **Entity** list, select **Carmakeent**.
 - e. In the **Case property** list, select **carMake**.
 - f. Click **Add mapping**.
 - g. In the **Entity** list, select **Carmodelent**.
 - h. In the **Case property** list, select **carModel**.
 - i. Click **Submit**.

Response configuration

Response Text analysis Entities extraction

Map entities types to **Insurance Quote** case model.

Entity	Case property
Carmakeent	carMake
Carmodelent	carModel

[+ Add mapping](#)

Cancel **Submit**

The Response configuration dialog box for case type commands

5. Build the model and refresh the updated entity information in the chatbot by clicking **Build model**.

Testing changes in the chatbot

After creating new entities, mapping the extraction of the entities to case properties, and rebuilding the model, you can test your changes to ensure that the chatbot responds correctly. In the preview console, verify that the chatbot automatically extracts the information about the car make and model that you mapped to the entities, and saves this information in the related Insurance Quote case properties.

When you provide the car make and model in a request for information about car insurance, the chatbot automatically skips the first two questions for which it recognized the answer and assigns the details to the Insurance Quote case properties.

1. In the preview console, click **Reset**.
2. Enter the same text as in step 1 ([on page 240](#)) of the *Configuring and mapping entities to case type properties* section to request an insurance quote.

Preview console

☐ Show analysis
 Reset

I would like to obtain an insurance quote for my Ford Mustang GT

#insurance

Start case - Insurance Quote - I-62

Please provide VIN number.

Type something to try...

Sample insurance quote

Tip: If the correct topic is already associated with your input, you can improve the chatbot's future responses in the model in the future by clicking the **Like** icon next to the topic name, and then rebuilding the model.

The chatbot correctly associates the text with the Insurance Quote case type, and starts a case. However, this time the bot asks only the third question about the VIN number of the car.

3. Display more information about the conversation and the case by turning on the **Show analysis** switch.

Preview console

Show analysis

Reset

Create case: Insurance Quote

1.953 sec

Neutral (0.0)

I would like to obtain an insurance quote for my

Ford

Mustang GT

Entity types

Carmakeent

Carmodelent

Topic

#insurance

Confidence

97%

Start case - Insurance Quote - I-62

What is the make of your car?

Ask (skipped): carMake

Value

Ford

Type

Carmakeent

What is the model of your car?

Ask (skipped): carModel

Value

Mustang GT

Type

Carmodelent

Please provide VIN number.

Type something to try...

The first and the second question about the make and model of the car are greyed out because they were skipped. The reason that they were skipped is that the *Carmakent* and *Carmodelent* entities were successfully extracted from your input, and automatically mapped to the *.carMake* and *.CarModel* case properties, respectively.

Conclusion

You made configuration changes to the chatbot algorithm, rebuilt the text analytics model, and successfully verified that the system improves how it responds to user requests for car insurance quotes. If the chatbot detects the correct topic and entities in user input, the system automatically skips the first two questions that it recognized the answer for, and then assigns the details to the Insurance Quote case properties.

Use case: Training and building the model in an IVA for Legacy Webchat

Improve how Pega Intelligent Virtual Assistant™ (IVA) for Legacy Webchat responds to requests by training the text analytics model. In this tutorial, you will learn how to create a sample training data and map entities, add the training data to the model, and when ready, rebuild the model. As a result, the chatbot improves how it responds to inquiries about a car insurance quote.

Note: Pegasystems will sunset the Legacy Webchat servers on October 31, 2023. Legacy Webchat has been replaced with the Web Messaging channel from Digital Messaging. Legacy Webchat was deprecated in 8.6 and de-supported in 8.7. Legacy Webchat is not supported for 8.7 and has not been tested. To learn how to migrate to Web Messaging, see [Migrating from Legacy Webchat to Web Messaging \(on page 130\)](#).

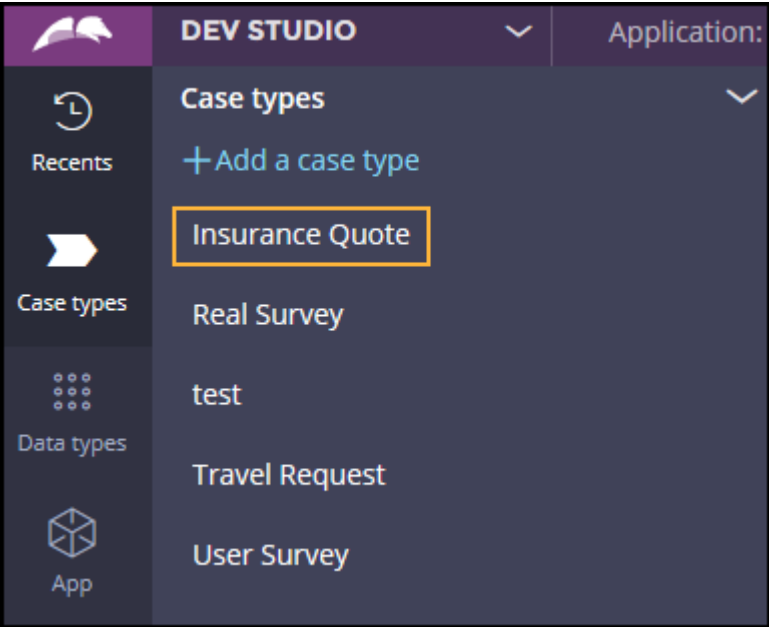
Although this content demonstrates how to train and build the model for the IVA for Legacy Webchat, you can perform similar steps to improve the model for the IVA for Alexa, Digital Messaging, and Pega Email Bot.

Before you begin

To follow the steps in this tutorial, set up a sample Pega Platform application with the Insurance Quote case type and a Legacy Webchat channel.

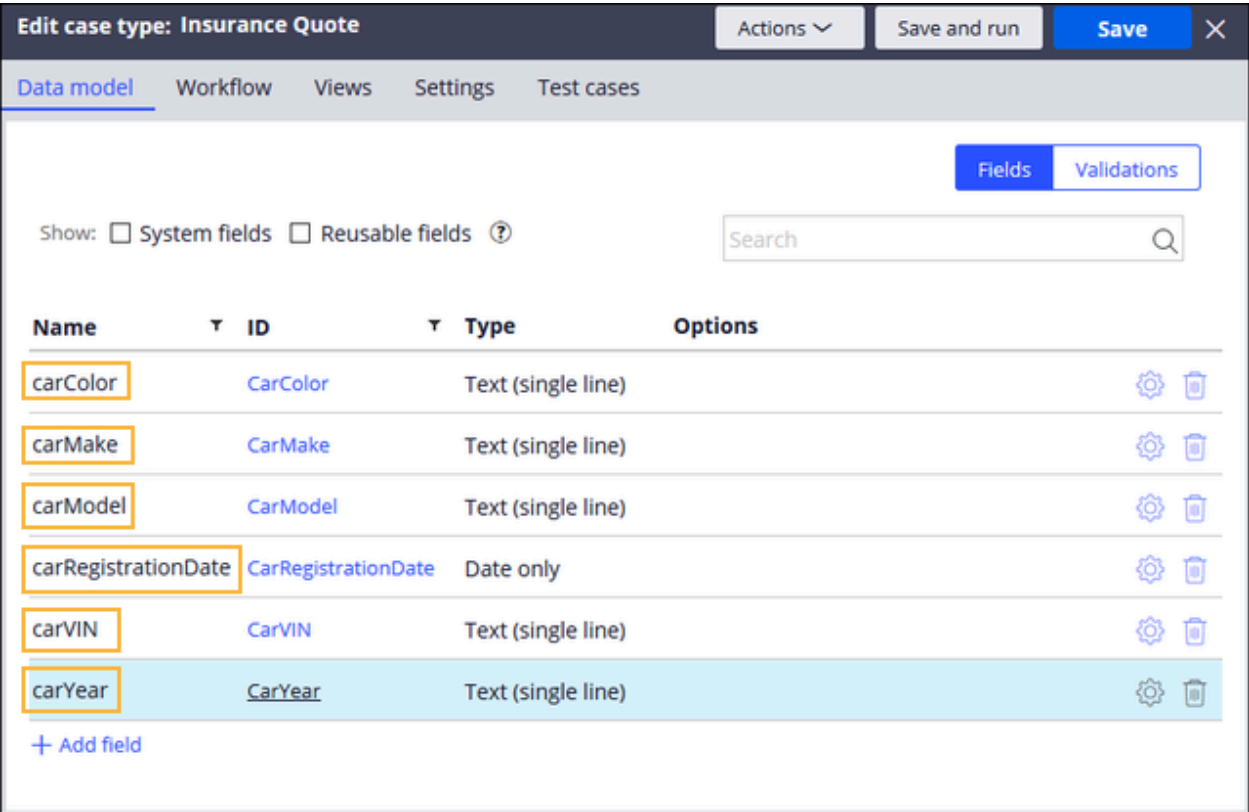
Set up the Insurance Quote case type for car insurance quotes:

1. Create the Insurance Quote case type.



Case type explorer

2. Add the following data types to the Insurance Quote case type for the car color, car make, model, car registration date, vehicle identification number (VIN), and car production year: *carColor*, *carMake*, *carRegistrationDate*, *carModel*, *carVIN*, and *carYear*.



Associating data objects with case types

For more information, see [Associating data objects with case types \(on page 248\)](#).

3. Enable Legacy Webchat channel-specific conversations for a stage of the Insurance Quote case type.

Stage

General Validation Goal & deadline

When all processes in this stage are complete:

☐ Automatically move to next stage

☒ Wait for a user action

☐ Resolve the case

Set case status on stage entry

Channel specific conversations

☒ WebChat

Stage configuration for the Insurance Quote case type

For more information, see [Adding a conversational channel to a case type process \(on page 180\)](#).

4. Configure conversation questions for a stage process of the Insurance Quote case type by adding three simple questions for the make, model, and VIN data types.

Conversation for WebChat-init

What is the make of your car? Text (single line)

What is the model of your car? Text (single line)

Please provide VIN number. Text (single line)

+ Add question

Sample questions for the Insurance Quote case type

5. Configure a Legacy Webchat channel with the following settings:

- In the **Content** section on the **Configuration** tab, add the *insurance* create case command for the Insurance Quote case type.

The screenshot shows the 'Content' configuration section. It contains two tables. The first table, 'Create case command', has columns for 'Create case command', 'Case type', and 'Authentication'. It lists three commands: 'insurance' (highlighted with an orange box) for 'Insurance Quote', 'travel' for 'Travel Request', and 'survey' for 'Real Survey'. Below this table is a '+ Add case type' button. The second table, 'Response command', has columns for 'Response command', 'Response type', and 'Authentication'. It lists two commands: 'hi' for 'Text' and 'cancel' for 'Cancel'. Below this table is a '+ Add response' button.

Create case command	Case type	Authentication
insurance	Insurance Quote	
travel	Travel Request	
survey	Real Survey	

+ Add case type

Response command	Response type	Authentication
hi	Text	Required
cancel	Cancel	

+ Add response

Sample create case command for a case type

For more information, see [Adding case commands for a conversational channel \(on page 181\)](#).

- On the **Behavior** tab, enable advanced text analyzer configuration and add the iNLP text analyzer. Ensure that the IVA can interpret text in both the context of the case and outside the case context.

The screenshot shows the 'Text Analyzer' configuration section. It has a checkbox 'Use advanced configuration' which is checked. Below this is a table with columns for 'Type', 'Apply in context of a case', and 'Apply outside of case context'. It lists two analyzers: '1 Exact match' and '2 Pega NLP generated' (highlighted with an orange box). The 'Pega NLP generated' analyzer has an 'Edit topics' button next to it. Below the table is a '+ Add text analyzer' button.

Text Analyzer ☒ Use advanced configuration
Text analyzers that process user's input and look for matching responses.

Type	Apply in context of a case	Apply outside of case context
1 Exact match	☒	☒
2 Pega NLP generated Edit topics	☒	☒

+ Add text analyzer

Enabled text analyzers

For more information, see [Adding a text analyzer for an IVA \(on page 188\)](#).

- Map the following entities to the Insurance Quote case data properties: `.carMake`, `.carModel`, and `.carVIN`.

Response configuration

Response
Text analysis
Entities extraction

Map entities types to **Insurance Quote** case model.

Entity		Case property		
Carmakeent	▼	carMake	▼	🗑️
Carmodelent	▼	carModel	▼	🗑️
Carvinent	▼	carVIN	▼	🗑️

[+ Add mapping](#)

Cancel
Submit

The Response configuration dialog box for case type commands

For more information, see [Mapping entities in conversation text \(on page 203\)](#).

In the preview console, verify that IVA for Legacy Webchat does not understand text that you enter to request a car insurance quote.

Preview console

☐ Show analysis Reset



Hi, I am a Pega Platform chatbot. How can I help you?

I need car insurance

Define topic



Sorry, I did not understand that

Car insurance

Define topic



Sorry, I did not understand that

I need quote

Define topic



Sorry, I did not understand that

I need car insurance for my Ford Mustang GT 1969

Define topic



Sorry, I did not understand that

Type something to try...

Sample chat conversation in the preview console

For more information, see [Verifying detected topics and entities \(on page 199\)](#).

Creating training records

Create training data records that contain sample user input for inquiries about a car insurance quote, so that you can later add this sample data to the model for the IVA. After you add enough instances of the same training data and build the model, the chatbot will improve its responses to user input about a car insurance quote. In addition, create five more instances of the same training data records in which you ask about an insurance quote for a Ford Mustang GT. This car make and model will later be mapped to entities, so that the system learns how to extract car make and model information and add it automatically to the Insurance Quote case type properties.

It is not sufficient to add only a few instances of the same training data. For each topic, create at least 20 records in the training sample so that the system learns how to correctly respond to the entered text. Distribute records evenly across topics, otherwise the model will be biased toward the overrepresented topic. For more information, see *Best practices for creating categorization models* (on page 254).

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Legacy Webchat channel that includes the Insurance Quote case type.
4. In the channel, click the **Training data** tab, and then click **Add records**.
5. In the **Create new training record** window, in the **Topic** field, press the Down arrow key and select: insurance
6. In the next field, enter the following sample user input inquiring about car insurance: I need a car quote.

Create new training record

Topic

insurance

I need a car quote

Close Create record

Creating a new sample training record

7. Click **Create record**.
8. Add more instances of the same training record by repeating steps [6 \(on page 253\)](#) through [7 \(on page 254\)](#) five more times.
9. In the next field, enter the following sample user input inquiring about insurance for a specific car make and model insurance: I need a car insurance quote for my Ford Mustang GT 2017.

Create new training record

Topic

insurance

I need a car insurance quote for my Ford Mustang GT 2017

Close Create record

Creating another sample training record

10. Click **Create record**.
11. Add more instances of the same data record by repeating steps 9 (on page 254) through 10 (on page 255) five more times.
12. Click **Close**.

Mapping entities in the training data

Map training data to entities so that the car make and model information is later automatically extracted from the text that you enter to the Insurance Quote case properties. After adding the training data and building the model, you will verify in the preview console that the car make and model from the user input is automatically copied to the Insurance Quote case properties.

In this example, you will associate the *#Carmakeent* and *#Carmodelent* entities with the car make and model in the user input. These entities are already mapped to the *.carMake* and *.carModel* properties that are defined for the Insurance Quote case.



Note: From the **Training data correction** window, you can also create entities to represent the user input.

1. On the **Training data** tab, select the training data record with the text: I need a car insurance quote for my Ford Mustang GT 2017
2. In the **Training data correction** window, select and right-click Ford, and then click **#Carmakeent**.
3. Select and right-click Mustang GT, and then click **#Carmodelent**.

Training data correction

Add to model
Edit

I need a car insurance quote for my Ford Mustang GT 2017

NLP analysis

Topic

insurance

Entities

Carmakeent
Carmodelent

Adding entities to sample training data

- Map more instances of the training data with the same text by repeating steps 1 (on page 255) through 3 (on page 255) five more times.

Adding training data to the model

Add sample training data to the model so that the IVA improves future responses to user inquiries about a car insurance quote. Update the model with sample data and associate it with the insurance topic. You can also help the system learn to recognize entities from user input such as the car make or model, so that this information is automatically mapped to the Insurance Quote case properties.



Note: Instead of using the **Training data** tab, you can also add sample training data and build the model directly by chatting with the chatbot in the preview console. For more information, see [Use case: Building an IVA for Legacy Webchat in the preview console \(on page 233\)](#).

1. On the **Training data** tab, select the checkboxes for the six instances of the text: I need a car quote

Configuration Behavior Connection **Training data** Mashup

Show topics : All

☒ 6 selected Delete Mark reviewed | Reviewed : 0 Add records

Manually added at 5/5/2020 6:55 AM	☒ I need a car quote
Manually added at 5/5/2020 6:55 AM	☒ I need a car quote
Manually added at 5/5/2020 6:55 AM	☒ I need a car quote
Manually added at 5/5/2020 6:55 AM	☒ I need a car quote
Manually added at 5/5/2020 6:55 AM	☒ I need a car quote
Manually added at 5/5/2020 6:55 AM	☒ I need a car quote

Sample training data records in Training data tab

2. Click **Mark reviewed**.
3. Select the checkboxes for the six instances of the text: I need a car insurance quote for my Ford Mustang GT 2017
4. Click **Mark reviewed**.

Building the model

Build the model with sufficient sample training data so that the IVA can recognize a user request for a car insurance quote. The system can then start the Insurance Quote case automatically, and ask for more information about the car make, model, and VIN.

1. On the **Training data** tab, click the **More** icon and then **Build model**.

You build the model by using only training data that was marked as reviewed.

Configuration Behavior Connection **Training data** Mashup

Model to train : My test channel
 Training data last added to the model at 5/5/2020 6:59 AM
 Model was last built at 5/5/2020 7:00 AM

Success:
 1. Topic model for English : Successfully updated the model performance from 1 to 1 F-score.

Show topics : All v

☐ Select All Delete Mark reviewed | Reviewed : 0 Add records ⋮

Manually added at 5/5/2020 6:59 AM ☐ I need car insurance for my Ford Mustang GT 1968
Manually added at 5/5/2020 6:59 AM ☐ Car insurance

Updating the model in the Training data tab.

The system successfully updates the model with the six training data records of each type that you created in the previous section.

2. Click **Save**.

Testing model changes in the chatbot

Verify the changes to the model for the IVA from the preview console. Enter the same training data that was used to train the model so that you can check that the system responds correctly to a request for a car insurance quote.

1. On the **Behavior** tab, in the preview console, enter the following text: I need a car quote.
2. Display more information about the conversation and the case by turning on the **Show analysis** switch.

Preview console

☒ Show analysis Reset

Hi, I am a Pega Platform chatbot. How can I help you?

I need a car quote
#insurance

Create case: Insurance Quote

0.086 sec Neutral (0.0)

I need a car quote

Topic	Confidence
#insurance	100%

Start case - Insurance Quote - I-122

What is the make of your car?

Ask: carMake

Type something to try...

Sample user input analysis inside a started case

The chatbot correctly responds, associating the text that you entered for the insurance topic with a confidence of 100 percent. The system also starts case I-122 and asks the first question about the make of the car.

3. Click **Reset**.
4. Enter the following text: I need a car insurance quote for my Ford Mustang GT 2017

Preview console

☒ Show analysis

Reset

I need a car insurance quote for my Ford Mustang GT 2017

#insurance Create case: Insurance Quote

2.544 sec

Neutral (0.0)

I need a car insurance quote for my **Ford**
Mustang GT 2017

Entity types

Carmakeent

Carmodelent

Topic

#insurance

Confidence

100%

Start case - Insurance Quote - I-123

What is the make of your car?

Ask (skipped): carMake

Value

Ford

Type

Carmakeent

What is the model of your car?

Ask (skipped): carModel

Value

Mustang GT

Type

Carmodelent

Pick a date

--

▼

--

▼

--

▼



The IVA correctly responds by associating the text that you entered with the insurance topic with a confidence of 100 percent. The system also starts case I-123. In addition, the car make and model that you entered, Ford and Mustang GT, are extracted and automatically mapped to the car make and model properties for the case. As a result, the IVA also skips the questions about the car make and model.

Conclusion

You have successfully added new training data to the model and rebuilt the model. As a result, you have trained and improved how the IVA responds to user requests about car insurance by using artificial intelligence and natural language processing (NLP).

Creating an Alexa channel

Create Pega Intelligent Virtual Assistant™ (IVA) for Alexa so that users interact with a Pega Platform™ application through voice commands to open a case or to perform other actions, such as displaying a menu of options.

Note: A chatbot works in one language because the conversational channel uses a template operator that is defined for a single locale. For more information, see [Template operator for IVA channel \(on page 171\)](#).

1. Create a new channel interface:
 - a. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
 - b. In the navigation pane of App Studio, click **Channels**.
 - c. In the **Create new channel interface** section, click **Alexa**.
 - d. In the **Details** section, in the **Channel interface name** field, enter a name for your channel.
 - e. **Optional:** To describe the purpose of the channel, in the **Description** field, enter a short description of your IVA.
 - f. In the channel, click the **Connection** tab.
 - g. In the **Template operator ID** list, click the name of an existing Pega Platform operator to use as a template.

The system uses this operator as a template to create all further accounts. Each account has a unique channel identifier set as its ID.

h. Click **Save**.

2. Create an Alexa skill definition.

An Amazon skill provides a built-in set of capabilities for Alexa, such as the ability to answer questions. Use the generated ID later, during chatbot configuration in Pega Platform. For more information, see [Creating an Alexa skill definition \(on page 263\)](#).

3. Set up an Alexa channel.

You set up the new Alexa channel in Pega Platform by using the settings that were generated in the Amazon developer portal. For more information, see [Setting up an Alexa channel \(on page 263\)](#).

4. Generate default skill configuration results and import them to the Amazon skill that you created in the Amazon developer portal.

For more information, see [Updating the Alexa skill configuration \(on page 264\)](#).

5. **Optional:** To test the Alexa skill definition, verify that the skill results work correctly.

For more information, see [Testing the Alexa skill definition \(on page 266\)](#).

[Template operator for IVA channel \(on page 171\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)

Installing the Alexa Channel component

To provide users with the Alexa channel to communicate with your application by voice commands through Amazon Alexa, add the *AlexaChannel* component to your Pega Platform application.

1. Switch to Dev Studio.

2. In Dev Studio, click the name of your application, and then click **Definition**.

3. In the **Enabled components** section, click **Manage components**.

4. If the *AlexaChannel* component does not appear in the list in the **Available components** section, install the **.zip** file on Pega Platform by clicking **Install new**.

The *AlexaChannel* component is not available in your application by default.

5. In the **Status** column, select the **Enabled** checkbox for the *AlexaChannel* component.

6. Click **OK**, and then click **Save**.

7. Switch back to App Studio.

[Creating an Alexa channel \(on page 261\)](#)

Creating an Alexa skill definition

To use Pega Intelligent Virtual Assistant™ (IVA) for Alexa with a Pega Platform application, add a new Amazon Alexa skill in the Amazon developer portal. An Amazon skill provides a built-in set of capabilities for Alexa so that IVA for Alexa can answer specific Pega Platform questions.

Creating an Amazon Alexa skill generates a unique application ID that you later use to configure the Alexa channel in Pega Platform to function as an IVA for Alexa.

1. From a web browser, log in to the Amazon developer portal.
2. Click the **Alexa** menu, and then create a new skill.
3. Click the **Skill name** tab, and then enter an invocation name for the customers to activate the Amazon skill.
4. From the **Default language** list, click **English (US)**.
5. In..., start a new skill definition by choosing the template.
6. Display a list of all Alexa skills, and then click **View Skill ID** for the new skill that you created.

A unique application ID for the new skill appears, for example,
`amzn1.ask.skill.62beb805-ae2b-493a-a15b-229131115ba8`

7. Save the unique application ID to a temporary file.

[Creating an Alexa channel \(on page 261\)](#)

Setting up an Alexa channel

Create Pega Intelligent Virtual Assistant™ (IVA) for Alexa so that users can interact with a Pega Platform application to ask questions, report an issue, or step through a process by using voice commands. For example, with an IVA for Alexa, users can create a case to inquire about a car price estimate or to book a flight.

The default chatbot skill configuration results include sample intent schema, a list of answers for all questions, and content for sample utterances that helps users order a pizza. The sample utterances also include unknown definitions for unknown responses. They force the Alexa chatbot to say "I don't understand" for responses that it does not understand.

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Create new channel interface** section, click the icon that represents your existing Alexa channel.
4. In the Alexa channel, click the **Connection** tab.



5. In the **Alexa application ID** field, enter the application ID for the Amazon Alexa skill that you saved to a file when you created an Alexa skill definition.
You specify the application ID value that was previously generated in the Amazon developer portal. For more information, see [Creating an Alexa skill definition \(on page 263\)](#).
6. Click **Save**, and then generate the default skill configuration results for the chatbot by clicking **Generate configuration**.
This information is later added to the Amazon Alexa skill definition that you created.
7. Save the information from the **Intent Schema** section to a temporary file.
8. Save the information from the **LIST_OF_ANSWERS_FOR_ALL_QUESTIONS** section to a temporary file.
9. Save the information from the **Sample Utterances** section to a temporary file.
10. **Optional:** If you update the chatbot case and response command content, click **Generate configuration** on the **Connection** tab, and then copy the generated content to the Amazon skill definition.

 **Note:** Do not modify the unknown responses.

[Creating an Alexa channel \(on page 261\)](#)

Updating the Alexa skill configuration

Update the information for the Alexa skill in the Amazon developer portal so that Pega Intelligent Virtual Assistant™ (IVA) for Amazon can communicate with the Alexa developer portal. You enter the skill configuration results that you generated for your IVA for Alexa, the URL for a Pega Platform instance, and the SSL certificate configuration.

1. In a web browser, log in to the Amazon developer portal.
2. From the list of Alexa skills, open the Alexa skill that you previously created.
3. Follow the instructions provided in the Alexa developer console to create and train the interaction model.
For more information, see the Alexa developer documentation related to the Alexa skills kit and skill development process.
4. **Optional:** In the **Slot Types** section, click **Add**, and then perform the following actions:
 - a. In the **Create custom slot type** field, enter: `LIST_OF_ANSWERS_FOR_ALL_QUESTIONS`
 - b. Click **Create custom slot type**, and then click **Bulk Edit**.
 - c. Copy the list of answers for all questions that you saved in a temporary file, and then paste the content into the field area.

For more information, see [Setting up an Alexa channel \(on page 263\)](#).

- d. Click **Submit**.
5. **Optional:** To set up a list of sample utterances, in the **Intents** section, click **Add**, and then perform the following actions:
 - a. In the **Create custom intent** field, enter a name for an intent.
 - b. Click **Create custom intent**, and then click **Build Edit**.
 - c. Copy the sample utterances that you saved in a temporary file in step, and then paste the content into the field area.
For more information, see [Setting up an Alexa channel \(on page 263\)](#).
 - d. Click **Submit**.
6. Click **Save Model**.
7. Select the certificate endpoints for your target customers:
 - a. Click the **Endpoint** section.
 - b. For the service endpoint type, click **HTTPS**.
 - c. In the **Default Region** field, enter the URL for your Pega Platform instance and its Alexa channel in the following format: `https://<PEGA_PLATFORM_INSTANCE>/PRRestService/alexachannel/srv/alexa`
 - d. Select the type of the certificate that your organization uses.
 - e. Click **Save Endpoints**.
8. Set up account linking information:
 - a. Click the **Account Linking** section.
 - b. Turn on the **Do you allow users to create an account or link to an existing account with you?** switch.
 - c. Click **Implicit Grant**.
 - d. Turn off the **Allow users to enable skill without account linking** switch.
 - e. In the **Authorization URI** field, enter the URL for your Pega Platform instance and its Alexa channel.
Use the same Pega Platform instance and the corresponding Alexa channel as in step [7.c \(on page 265\)](#).



f. In the **Client ID** field, enter the application ID for the Alexa skill that you saved to a file when you created an Alexa skill definition.

For more information, see [Creating an Alexa skill definition \(on page 263\)](#).

g. Click **Save**.

[Creating an Alexa channel \(on page 261\)](#)

Testing the Alexa skill definition

After you configure an Alexa skill with the Alexa channel information, you can test whether the configured skill results work correctly. You can run the tests in the Amazon developer portal.

1. In a web browser, log in to the Amazon developer portal.
2. On the **Alexa** menu, in the **Alexa Skills Kit** section, click **Get Started**, and then click the name of the your Alexa skill.
3. Click the **Test** tab.
4. Turn the switch to **Enabled**.
5. In the **Service Simulator** section, click the **Test** tab.
6. In the **Enter Utterance** field, enter a sample command for Alexa, and then click **Ask <Alexa_Skill>**.
The **Service Request** and **Service Response** fields display your request and the generated response from Alexa in JSON format.
7. **Optional:** To listen to the generated Alexa response, click the **Listen** icon .
8. Repeat steps [6 \(on page 266\)](#) and [7 \(on page 266\)](#) to test additional commands for Alexa.

[Creating an Alexa channel \(on page 261\)](#)

Creating an IVA channel from an existing channel

Save time building your chatbot by creating Pega Intelligent Virtual Assistant™ (IVA) based on a verified configuration in another IVA channel.

For example, in your production environment, if you have an IVA for Digital Messaging that intelligently responds to customer inquiries, starts business cases, and interacts with your Pega Platform™ application, you can reuse that Digital Messaging channel configuration in a new channel.

When you create a Digital Messaging or Legacy Webchat channel based on a configuration in another IVA channel, the system copies the following information from the existing channel:

- Conversation flows
- Case commands
- Response commands



- Event commands
- Text analyzers
- Settings in the **Options** section
- Template operator

1. In the header of Dev Studio, click the name of the application, and then click **Channels and interfaces**.
2. In the navigation pane of App Studio, click **Channels**.
3. In the **Current channel interfaces** section, click the icon that represents your existing Digital Messaging or Legacy Webchat channel with the settings that you want to copy to the new channel.
4. In the upper right of the page, click **Actions > Copy channel interface**.
5. In the **Copy channel interface** window, click the Digital Messaging or Legacy Webchat icon that represents the channel that you want to create.

The system displays the new channel.

6. In the **Details** section, in the **Channel interface name** field, enter a name for your channel.
7. **Optional:** To describe the purpose of the channel, in the **Description** field, enter a short description of your IVA.
8. For Digital Messaging channels, click the **Connection** tab, and then, in the **Base URL** field, enter a new URL for your system.
9. Click **Save**.

[Digital Messaging channel overview \(on page 116\)](#)

[Implementing an Intelligent Virtual Assistant for a conversational channel \(on page 112\)](#)