

Pega Sales Automation Implementation Guide 8.4



©2020 Pegasystems Inc., Cambridge, MA. All rights reserved.

Trademarks

For Pegasystems Inc. trademarks and registered trademarks, all rights reserved. All other trademarks or service marks are property of their respective holders.

For information about the third-party software that is delivered with the product, refer to the third-party license file on your installation media that is specific to your release.

Notices

This publication describes and/or represents products and services of Pegasystems Inc. It may contain trade secrets and proprietary information that are protected by various federal, state, and international laws, and distributed under licenses restricting their use, copying, modification, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

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

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

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors, as well as technical inaccuracies. Pegasystems Inc. shall not be liable for technical or editorial errors or omissions contained herein. Pegasystems Inc. may make improvements and/or changes to the publication at any time without notice.

Any references in this publication to non-Pegasystems websites are provided for convenience only and do not serve as an endorsement of these websites. The materials at these websites are not part of the material for Pegasystems products, and use of those websites is at your own risk.

Information concerning non-Pegasystems products was obtained from the suppliers of those products, their publications, or other publicly available sources. Address questions about non-Pegasystems products to the suppliers of those products.

This publication may contain examples used in daily business operations that include the names of people, companies, products, and other third-party publications. Such examples are fictitious and any similarity to the names or other data used by an actual business enterprise or individual is coincidental.

This document is the property of:

Pegasystems Inc.
One Rogers Street
Cambridge, MA 02142-1209, USA
Phone: 617-374-9600 Fax: 617-374-9620

www.pegasystems.com

Document: Pega Sales Automation Implementation Guide

Publication date: February 25, 2020

Feedback

If you have comments for how we can improve our materials, send an email to AppDocBug@pegasystems.com.

Contents

Overview.....	6
Purpose of this guide.....	6
Pega Sales Automation.....	6
Application stack.....	7
Preparing for the implementation.....	8
Creating the application.....	8
Optional: Configuring your Pega applications to support CRM.....	9
Adding shared rulesets.....	10
Creating integration rulesets.....	11
Enabling shared features.....	12
Configuring the integration between Pega Sales Automation and Pega Marketing.....	12
Testing the Pega Sales Automation integration with Pega Marketing local campaigns.....	13
Testing the Pega Sales Automation integration with Pega Marketing next best offers.....	13
Building features.....	14
Implementing the application data model.....	14
Mapping your application data.....	14
Managing your application data with the Integration Designer.....	15
Mapping case type or data type classes to a database table.....	15
Connecting to your sales data.....	15
Configuring the sales team structure.....	16
Sales team.....	16
Importing existing sales team data.....	17
Adding territories.....	17
Adding operators.....	18
Adding partners.....	18
Configuring prospect and customer data.....	19
Prospect and customer data types.....	19
Importing prospect and customer data.....	20
Configuring data loaders extensions.....	20
Opportunity splits between sales team members.....	21
Configuring opportunity splits.....	21
Enabling application features.....	22
Activating application features.....	22
Implementing the Sales methodology.....	23
Defining the sales methodology.....	24
Lead and opportunity case types.....	24
Configuring the sales methodology.....	25
Implementing case types.....	25
Modifying an opportunity case type.....	25
Adding stages to an opportunity case type.....	25
Specifying a field value for a new opportunity stage.....	26
Adding steps to an opportunity stage.....	26
Implementing products and competitors.....	26
Defining products and competitors.....	27
Setting up products.....	27

Configuring an external product data source.....	28
Setting up competitors.....	28
Implementing forecast management requirements.....	28
Implementing activity and task management requirements.....	28
Implementing sales goals.....	29
Defining requirements for sales goals management.....	29
Setting up time periods for sales goals management.....	29
Implementing reporting.....	29
Defining reporting requirements.....	30
Reviewing existing reports.....	30
Reports.....	30
Implementing a mobile app.....	30
Completing prerequisite tasks before building the Pega Sales Automation mobile app.....	30
Building the Pega Sales Automation mobile app.....	31
Reviewing app features in the sample application.....	32
Reviewing rules and patterns for reuse in the mobile app.....	32
Deploying your mobile app.....	32
Setting up optical character recognition (OCR) scanning.....	33
Implementing Marketing features.....	33
Local campaigns.....	33
Next-best-offer recommendations.....	34
Enabling local campaigns and next-best-offer recommendations.....	34
Next-best-action strategies.....	34
Implementing engagement maps.....	35
Implementing organization trend data.....	35
Defining organization trend requirements.....	35
Mapping data for organization trends.....	35
Implementing Pega Co-Browse.....	36
Defining Pega Co-Browse requirements.....	36
Configuring Pega Co-Browse.....	36
Configuring tagging for work objects and Pulse posts.....	37
Configuring Pulse.....	38
Implementing Pega Call.....	38
Defining computer telephony integration (CTI) requirements with Pega Call.....	38
Configuring computer telephony integration using Pega Call.....	39
Implementing Pega Knowledge.....	39
Pega Knowledge integration.....	39
Adding Pega Knowledge records.....	40
Implementing artificial intelligence.....	40
Pega Workforce Intelligence integration.....	40
Configuring Workforce Intelligence.....	41
Configuring settings for Workforce Intelligence.....	41
Configuring authorization for Workforce Intelligence.....	41
Updating the access group for the agent schedule.....	42
Configuring Pega Sales Automation operators for Workforce Intelligence.....	42
Implementing third-party integrations.....	42
Defining integration requirements.....	43
Pega Sales Automation integration with Microsoft Exchange.....	43
Configuring Pega for Outlook VSTO and Pega for Outlook Office.....	43
Configuring listener accounts for emails and appointments.....	44
Creating the access group, operator, and email accounts.....	44
Creating new email and appointment classes in the implementation layer.....	44
Configuring associations with other application objects.....	45
Updating the user interface.....	45
Configuring Twitter integration.....	46
Setting up a Twitter account and application.....	46

Configuring settings for Twitter integration.....	46
Configuring Facebook integration.....	47
Setting up a Facebook account and application.....	47
Configuring settings for Facebook integration.....	47
Configuring LinkedIn integration.....	48
Configuring Partner Relationship Management.....	48
Configuring deal registration.....	51
Configuring Market Development Fund (MDF).....	52
Implementing the security model.....	52
Authentication schemes.....	52
Implementing your authentication scheme.....	53
Authorization model.....	53
Operator privileges by role.....	53
Work object permissions.....	54
Attribute Based Control (ABAC).....	54
Client-based access control (CBAC).....	55
Modifying the user interface.....	55
Designing for screen performance.....	55
Implementing dashboards.....	55
Determining which part of the user interface to modify.....	55
Portal dashboards.....	56
Defining the dashboard layouts for portals.....	56
Customizing dashboards.....	56
Reviewing the content of the portals.....	57
Locating the user interface element to change.....	57
Adding properties to sections.....	57
Adding sections.....	58
Changing section visibility.....	58
Designing an application skin and styles.....	58
Updating the skin.....	58
Testing a new application.....	60
Testing your application in the Build environment.....	60
Testing in the test or production environments.....	60
Testing in the UAT environment.....	61
Packaging a new application.....	62
Merging application changes.....	62
Packaging an application for migration.....	62
Importing the packaged application.....	62
Production maintenance and monitoring.....	63
Business rule maintenance in the Production environment.....	63
Application health monitoring.....	63
Identifying and reporting issues.....	63

Overview

Before implementing Pega Sales Automation, it's helpful to understand how to use this guide, the application business uses, and the application stack.

- [Purpose of this guide](#)
- [Pega Sales Automation](#)
- [Application stack](#)

Purpose of this guide

A project team that implements a Pega application uses this implementation guide to enable and extend application features to meet client business requirements. Complete the product installation before starting the implementation procedures. For information about installing your application, see the product installation guides on the [Pega Sales Automation product page](#).

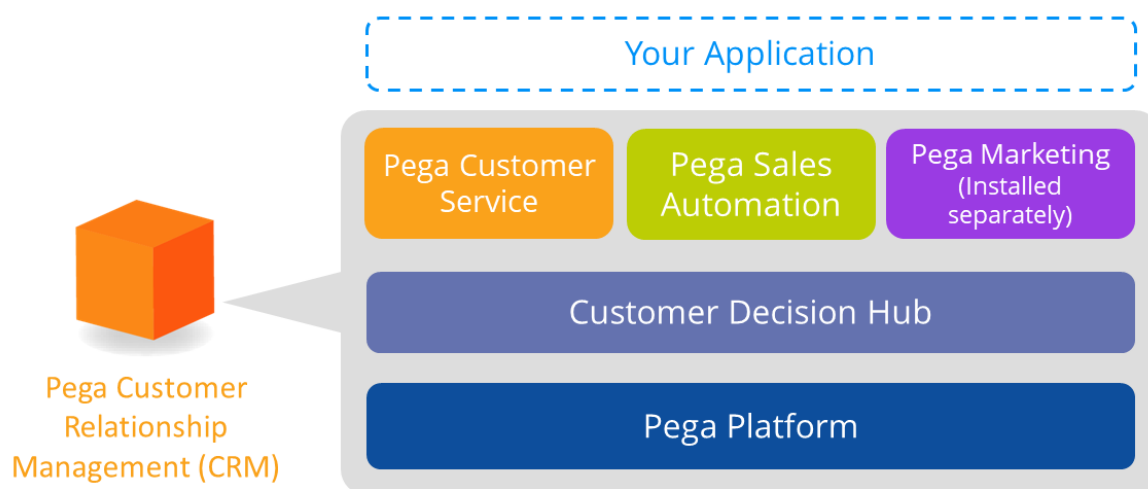
This implementation guide presents tasks in the sequence in which they are commonly performed; however, the sequence varies based on business priorities. Follow the procedures in this guide to complete the tasks for the first minimum lovable product (MLP) release. Also, use this guide during iterative releases as you configure and extend more features on top of the MLP release.

The application includes a case type backlog that supports the procedures in this implementation guide. The case type backlog is an inventory of the application-provided case types that establishes the project scope by defining the changes and additions that are required for the first MLP release and later releases. For more information, see [Capture microjourneys™ in the case type backlog](#) and [Journey Centric Delivery](#).

Pega Sales Automation

Pega Sales Automation combines analytics, business process management, and model-driven capabilities to provide intelligent, real-time sales guidance, process automation from lead to fulfillment, and the agility to quickly adapt existing sales processes to keep up with changes in the market. With Pega Sales Automation, you can standardize and optimize the sales process across multiple business units, products, team members, territories, accounts, and channels, and dynamically adapt to each specialized circumstance.

Application stack



A new Pega Sales Automation installation includes Pega Customer Service, but Pega Customer Service is enabled only if licensed. If you are using Pega Marketing, you must install that application separately.

Preparing for the implementation

To prepare the implementation environment and to create your application, complete the preparation tasks.

- [Creating the application](#)
- [Optional: Configuring your Pega applications to support CRM](#)

Creating the application

Run the New Application wizard to create your application.

1. To create a new operator ID for running the New Application wizard, complete the following steps:
 - a. Log in to Dev Studio by using the operator ID `administrator@pega.com` and the password that you specified for that operator.
 - b. Save a copy of the existing `administrator@pega.com` operator, and give it a name that identifies it as an Application Setup operator.
 - c. Add the *PegaCRM-SFA:AppSetup* access group to the new operator record, and then click the radio button to the left of the access group to select it as the default access group.
When this access group is selected as the default access group for an operator, the New Application wizard opens when the operator logs in.
 - d. Save the new Application Setup operator.
2. Log in as the Application Setup operator.
3. Follow the New Application wizard instructions until the **Name your application** page opens, and then follow the steps below.

For more information about each step of the wizard, see [Creating an application](#).

4. On the **Name your application** page, enter the name of the application, and then click **Advanced configuration**.
5. In the **Organization settings** section, enter the **Organization name**, **Division name**, and **Unit name** for this application.

The New Application wizard creates the application class structure for you based on the organization settings that you enter. For more information, see [Class layers](#) and [class hierarchy and inheritance](#).

If you have not already defined the organization entities (for example, if you have not already defined the division), type the name of the new entity in the appropriate field. The application saves the new values when you create the new application.

6. Click **Save**.
7. Click **Create application**.

The Application Wizard creates the implementation application. The application includes one system administrator operator so that you can log into the application after you complete the wizard.

8. To open the new application, click **Go to app**.

The New Application wizard creates a set of access groups for the application. Create your own operators, and then apply the appropriate access groups.

What to do next: After you create your application, enable and extend configurable functionality and features to meet your business needs.

Optional: Configuring your Pega applications to support CRM

These procedures are required only if you will use the [Pega Customer Relationship Management features](#) to share data and service cases across the Pega Customer Service, Pega Sales Automation, and Pega Marketing applications. These procedures involve configuration across multiple applications, and therefore include links to multiple application implementation guides.

Before you begin:

Your Pega Customer Service, Pega Sales Automation, and Pega Marketing applications must already be created as described in [Creating your application](#).

If you have installed industry-specific applications on top of Pega Customer Service and Pega Sales Automation, complete these configuration steps in the industry applications, and not in the Pega Customer Service or Pega Sales Automation applications. For instructions, see the implementation guide for the industry application on the [Pega Community](#).

To configure your Pega applications to support shared features, complete the following steps:

1. For environments that include both Pega Customer Service and Pega Sales Automation, complete the following steps:
 - a. Add the Pega Sales Automation implementation rulesets to your Pega Customer Service application, and add the Pega Customer Service implementation rulesets to your Pega Sales Automation application. See [Adding shared rulesets](#).
 - b. Create an integration ruleset for your Pega Customer Service and Pega Sales Automation applications, and then add the integration ruleset to the top of the ruleset stack for each application. See [Creating integration rulesets](#).
 - c. Enable the Pega Customer Relationship Management features.
See [Enabling shared features](#).
 - d. In your Pega Customer Service application, update the operator access groups to include the `CustomerService:CSRSalesAccess` role that is required to access the shared features.
See "Updating access groups to have shared access" in the *Pega Customer Service Implementation Guide* on the [Pega Customer Service product page](#).
2. For environments that also include Pega Marketing, complete these additional steps.
 - a. Configure your Pega Customer Service application to support Pega Customer Decision Hub.
 1. Enable Pega Customer Decision Hub. See "Configuring application settings" in the *Pega Customer Service Implementation Guide* on the [Pega Customer Service product page](#).
 2. Configure Customer Decision Hub settings. For each setting, copy the rule to the highest ruleset shared by Pega Customer Service and Pega Sales Automation, and save your changes in that ruleset. See "Configuring Pega Customer Decision Hub" in the *Pega Customer Service Implementation Guide* on the [Pega Customer Service product page](#).

- b. In Pega Marketing, configure the application to support the Pega Customer Relationship Management features.
See "Configuring Pega Marketing to support the Pega Customer Relationship Management features" on the *Pega Marketing Implementation Guide* on the [Pega Marketing product page](#).
- c. In your Pega Sales Automation application, configure the integration between Pega Sales Automation and Pega Marketing.
See [Configuring the integration between Pega Sales Automation and Pega Marketing](#).
- d. In your Pega Sales Automation application, test the integration with Pega Marketing local campaigns.
See [Testing the integration with local campaigns](#).
- e. In Pega Marketing, train the adaptive model for sales offers.
See "Configuring Adaptive Decision Manager for sales offers" in the *Pega Marketing Implementation Guide* on the [Pega Marketing product page](#).
- f. In Pega Marketing, run a Marketing campaign for batch evaluation.
See "Running a marketing campaign for batch evaluation" in the *Pega Marketing Implementation Guide* on the [Pega Marketing product page](#).
- g. In your Pega Sales Automation application, test the integration with Pega Marketing next best offers. See [Testing the integration with next best offers](#).

- [Adding shared rulesets](#)
- [Creating integration rulesets](#)
- [Enabling shared features](#)
- [Configuring the integration between Pega Sales Automation and Pega Marketing](#)
- [Testing the Pega Sales Automation integration with Pega Marketing local campaigns](#)
- [Testing the Pega Sales Automation integration with Pega Marketing next best offers](#)

Adding shared rulesets

To enable sharing of data between your Pega Customer Service and Pega Sales Automation applications, add the implementation rulesets from one application to the other. The implementation rulesets are created by the New Application wizard for a new application.

This procedure is required only if you are sharing data and service requests between two or more applications in the Pega Customer Relationship Management suite.

Add only the rulesets that you want to expose in the other application. The lead system architect (LSA) or project architect can provide guidance on what functionality is included in each ruleset.

Add these rulesets at the bottom of the ruleset stack so that you do not override your application's implementation rulesets. When you want to overwrite your application's rulesets, use an integration ruleset as described in [Creating integration rulesets](#).

1. View the implementation rulesets created for each of your applications.
 - a. Log in to each of your applications in Dev Studio.
 - b. Click **Application menu** > **Definition**.
 - c. In the **Application rulesets** section, note the implementation rulesets created for each application.
2. Add the implementation rulesets from your Pega Sales Automation application to the bottom of the ruleset stack for your Pega Customer Service application.

- a. Log in to your Pega Customer Service application in Dev Studio.
- b. Click **Application menu > Definition**.
- c. In the **Application rulesets** section, click **Add ruleset** to add Pega Sales Automation rulesets to the bottom of the ruleset stack, and then click **Save**.
3. Add the implementation rulesets from your Pega Customer Service application to the bottom of the ruleset stack for the Pega Sales Automation application.
 - a. Log in to the Pega Sales Automation implementation application in Dev Studio.
 - b. Click **Application menu > Definition**.
 - c. In the **Application rulesets** section, click **Add ruleset** to add Pega Customer Service rulesets to the bottom of the ruleset stack, and then click **Save**.

Creating integration rulesets

When using a Pega Customer Relationship Management configuration that includes both Pega Customer Service and Pega Sales Automation, you must sometimes override the rules in one application to expose functionality from the other application. To support this, create an integration ruleset in both applications, and then save the override rules to the integration ruleset.

For example, if the Pega Customer Service application will show Pega Sales Automation *Lead case* service cases, you want the service case header and footer to look the same as the other Pega Customer Service services cases. In that case, save the perform harness rule for the service case into the integration ruleset for the Pega Customer Service application. Doing so overrides the Pega Sales Automation header and footer for that service case.

This procedure is required only if you are sharing data and service requests between two or more applications in the Pega Customer Relationship Management suite.

The applications include a sample integration ruleset that you can copy or use as a model for a new integration ruleset. The sample integration ruleset for Pega Customer Service is SAforCS, and the sample integration ruleset for Pega Sales Automation is CSforSA.

Complete the following steps to configure the integration rulesets:

1. Log in to your Pega Customer Service application in Dev Studio.
2. Click **Application menu > Definition**.
3. In the **Application rulesets** section, click **Add ruleset**.
4. Type a name for the new integration ruleset that identifies its purpose. For example, *MyApp* CRMIntegration, where *MyApp* is your application prefix.
5. Click the **Open** icon to the right of the ruleset field.
6. Review the default **Ruleset Name**, **Version**, and **Description** values, and modify the values if needed.
7. Click **Create and Open**, and then click **Save**.
8. In the ruleset field, identify which version of the ruleset to use by appending the major and minor rule version to the end of the integration ruleset name in this format: *:NN-NN*. For example, to select the first version of the ruleset, append *:01-01* to the end of the ruleset name.
9. Move the new integration ruleset to the top of the ruleset stack by dragging the number to the left of the ruleset name to the top of the stack.
10. Click **Save**.
11. Repeat this procedure for your Pega Sales Automation application.

Enabling shared features

To share data and service cases across the Pega Customer Service and Pega Sales Automation, enable the shared features. For more information about the shared features, see [Pega Customer Relationship Management suite of applications](#).

1. Open your Pega Customer Service application.
2. In the navigation pane of App Studio, click **Settings** > **General**.
3. In the list of general settings, click **Sales Automation**.
4. To enable your application to use the pre-integrated Pega Sales Automation features, select the **Enable Sales Automation** check box.
5. Click **Save**.
6. Open your Pega Sales Automation application.
7. In the navigation pane of App Studio, click **Settings** > **Application settings**.
8. Select the **Enable CRM mode** check box.
9. Click **Save**.

Configuring the integration between Pega Sales Automation and Pega Marketing

The Pega Customer Relationship Management suite includes sample marketing artifacts that you can import to demonstrate the next-best-offer and local campaigns features in Pega Sales Automation.

To use these artifacts in Pega Sales Automation you must configure the integration between Pega Sales Automation and Pega Marketing.

For more information about the next-best-action and local campaigns features, see [Implementing Marketing features](#).

For information about the integration steps you must complete in the Pega Marketing application, see [Optional: Configuring your Pega applications to support CRM](#).

1. In the navigation panel of Dev Studio, click **Records** > **SysAdmin** > **Remote system**.
2. Open the **NBAA** system record and configure it the following way:
 - a. On the **Environment** tab, select the **Connect using mashup** connection type.
 - b. In the **System URL** field, enter the URL of your Pega Marketing.
 - c. In the **Application Name** field, enter the name of your Pega Marketing application at the implementation layer.
3. In App Studio, click **Settings** > **Application Settings** and select the **Local campaigns** check box on the **Pega Marketing** tab.
4. Log out of the application and log back in.

Testing the Pega Sales Automation integration with Pega Marketing local campaigns

After you complete the steps required to configure the integration between Pega Sales Automation and Pega Marketing, verify that you can access the sample marketing artifacts for local campaigns in Pega Sales Automation.

1. Log in to Pega Sales Automation by entering the credentials for the Sales Rep operator who has `PegaSA:MarketReps` as their secondary access group.
2. In the header of Dev Studio, click **Launch web interface > SA for Sales Rep**.
3. On the navigation menu, click **Campaigns > New Campaign**.
4. Verify that you can select the following template offers for the campaign:
 - B2CEmailOffer
 - B2BEmailOffer
 - B2BSMSOffer
 - B2CSMSOffer
5. Verify that you can add Pega Sales Automation contacts to the campaign.

Testing the Pega Sales Automation integration with Pega Marketing next best offers

After you complete the steps required to configure the integration between Pega Sales Automation and Pega Marketing, verify that you can access the sample marketing artifacts for next-best-offer opportunities in Pega Sales Automation.

1. Log in to Pega Sales Automation by entering the credentials for the Sales Rep operator who has `PegaSA:MarketReps` as their secondary access group.
2. In the header of Dev Studio, click **Launch web interface > SA for Sales Rep**.
3. Verify that the Next-best-offer widget displays on the dashboard. With sample data uploaded, the widget shows realistic offers with different rankings for each contact.
4. Complete the following steps to test your integration:
 - a. Click a contact to qualify the contact for an opportunity.
 - b. Open a contact to display the top three offers for the contact, ranked highest to lowest.
 - c. Click an offer to qualify the contact for an opportunity to receive the offer.

Building features

The project team enables and extends features to meet your business requirements.

- [Implementing the application data model](#)
- [Connecting to your sales data](#)
- [Enabling application features](#)
- [Modifying the user interface](#)

Implementing the application data model

The application provides a set of data types, data pages, and sample data to begin implementing your application.

Data modeling involves relating a conceptual model of how data items relate to each other in an application. The data model in the Pega Platform refers to a set of rules that work together to populate the data in your application. This data is displayed on the user interface to help the user process information and can help automate decisions in your sales processes. The following rule types constitute your data model:

Data types

The data type is another name for a class in your application. It holds data that the application uses. A data type has one or more data pages and several property definitions associated with it.

Properties

Properties define the format and visual presentation of data in your application.

Data pages

Data pages define the content of a clipboard page and control the loading of the data.

When planning your data model, work with the data modeling resource at your site to understand the attributes of the data types that need to be supported by your application.



Note: For more information about the data model, see the [Pega Sales Automation Product Overview](#) on the [Pega Sales Automation product page](#).

The following topics describe how to plan and map the data model:

- [Mapping your application data](#)
- [Managing your application data with the Integration Designer](#)
- [Mapping case type or data type classes to a database table](#)

Mapping your application data

The application data types provide a default set of properties for your use. You can add or remove properties from these data types depending on your requirements. You can also add new data types.

1. In the navigation pane of Dev Studio, click **Data types**.
2. Click the data type that you want to modify.
3. On the **Data model** tab, determine the properties to add, remove, or leave unchanged.

4. Repeat these steps for any other data types, as necessary.

Managing your application data with the Integration Designer

Use the Integration Designer to manage data at the application level. In the Integration Designer, you can access a comprehensive view of all business objects in your application and their associated data, such as references, data sources, and data views. You can easily create and modify data objects and connect to data sources.

1. In the navigation pane of App Studio, click **Data**.
2. Locate each data type that is marked **SIMULATED**, and click the **FIX NOW** link to configure that data type to connect to real data.

For more information, see [Managing data and integrations with the Integration Designer](#).

Mapping case type or data type classes to a database table

When you run the New Application wizard to create your initial case types and data types, the system creates a default class group mapping for each case type and data type. The wizard also creates associated database tables. The wizard maps the new implementation class to the appropriate database table.

Mapping a class to a physical database table affects data storage and security decisions. For example, if you expect the volume of the case types to be low, consider sharing the table with other case types.

To update the default database table to a user-specified database table, perform the following steps:

1. In the navigation pane of Dev Studio, click **App**.
2. Expand the **SysAdmin** category, and then click **Database Table**.
3. Open the class that you want to modify.
4. Review the default database name and the table name that the class points to.
5. To point the chosen database table to your specific class, edit the database name and table name.
6. To confirm your changes, click **Save**.
7. Click **Test connectivity** to check if the configuration succeeded. If the test results display errors, for example, missing properties, review the data and correct them.
8. Update the class to point to the table specified above:
 - a. In the navigation pane of Dev Studio, click **App**.
 - b. Right-click the class name that you referred to in the steps above and select **Definition**.
 - c. On the **General** tab, click **Test connection** to check if the configuration succeeded.
9. Perform the steps above for the other case type or data type classes, if needed.

Connecting to your sales data

You start building your application by setting up the sales team, configuring organizational data, and importing prospect and customer data.



Note: Whether you import data or enter it manually, the order in which you enter data in the application is important. Setting up your sales team and configuring organizational data, such

as product hierarchies and competitor information, before importing or creating leads and opportunities ensures that the appropriate data components are automatically associated with the work objects in your application.

For information and instructions on how to build your sales team and import prospect and customer data, see the following topics:

- [Configuring the sales team structure](#)
- [Configuring prospect and customer data](#)
- [Configuring data loaders extensions](#)
- [Opportunity splits between sales team members](#)

Configuring the sales team structure

Configure the organizational structure of your sales team by setting up the territories, operators, and partners required for your business.

When you configure the sales team structure, you are also configuring your security authorization model because that model is based on the territory structure, operator IDs, and partners that you define for your sales team. For more information, see [Authorization model](#).

To set up the sales team, you can import existing entities as described in [Pega Sales Automation data import](#), and you can manually add entities.

- [Sales team](#)
- [Importing existing sales team data](#)
- [Adding territories](#)
- [Adding operators](#)
- [Adding partners](#)

Sales team

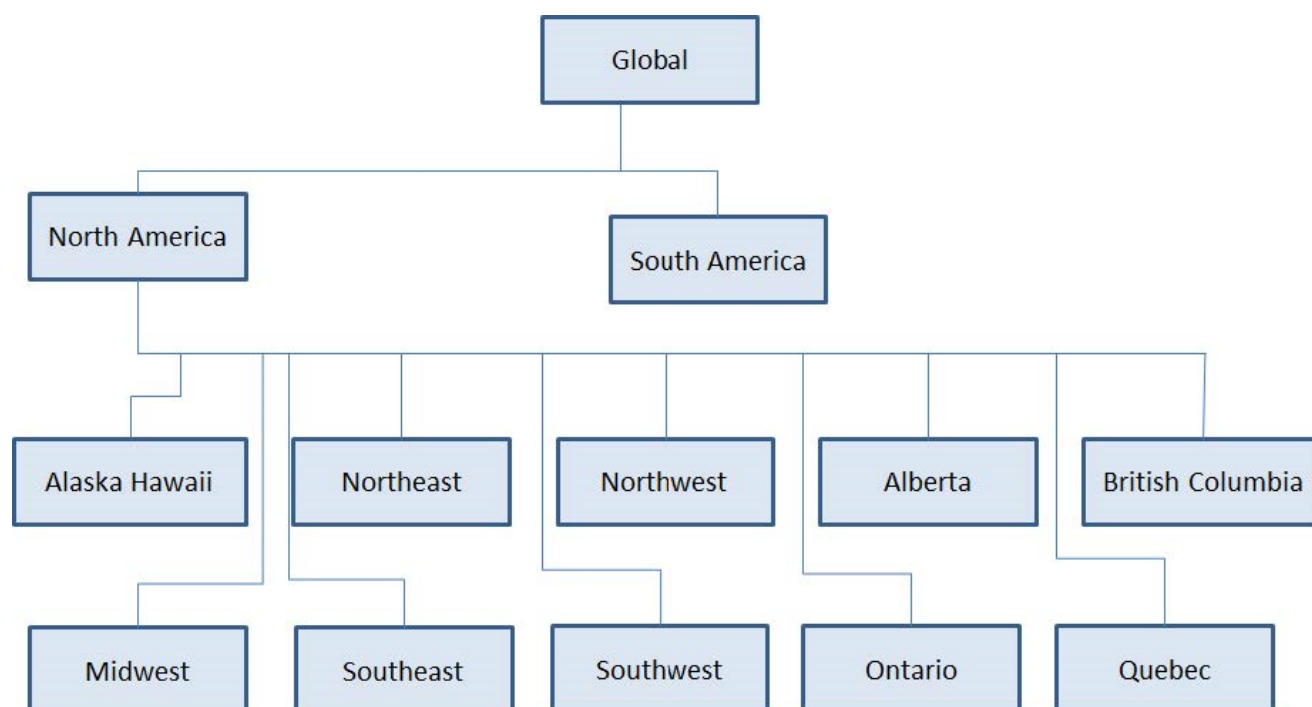
In Pega Sales Automation, the territories, operators, and partners that are part of the sales team have the following functions.

Territories

A territory is a hierarchical unit that is used to organize your sales team into groups, for example, a geographic location, a department, or a product line.

Assign operators to territories, and configure each operator's read, update, and create permissions (at the territory level) for the work objects that are used in the Pega Sales Automation application, including: accounts, contacts, leads, opportunities, organizations, and households.

The following diagram shows a territory structure based on geographic locations:



When setting up territories for a new application, set up territories before you set up operators. After you set up operators, you can edit a territory to optionally select which operator is the forecast manager.

Operators

An operator is an individual user of the Pega Sales Automation application, either a member of your sales team or sales agents at partner organizations. You define the sales team operators in the SA for Sales Ops portal. (These operators have a different purpose than the operators that you define in Dev Studio.)

Partners

With Pega Sales Automation, you can work directly with by granting them access to your local implementation.

The partner management feature is optional and you must activate it for your implementation if you plan to use it. For more information, see [Activating application features](#).

Importing existing sales team data

If you have existing sales team data set up in a different system, you can import the sales team data to Pega Sales Automation instead of adding that information manually.

1. Identify the types sales team data to import.
2. Import the sales team data using one of the methods described in [Pega Sales Automation data import](#).

Adding territories

Configure the territories that establish the structure for managing your sales team. For more information about territories, see [Sales team](#).

A Global Sales Ops operator can view and edit the entire territory tree structure. A Local Sales Ops operator can view and edit their primary territory and any children under that territory.

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.

2. In the navigation pane, click **Territories**.
3. Click **Create territory**.
4. Complete the required fields and enter optional information as needed for your implementation:
 - a. **Optional:**
Optional: If you are setting up a territory for one of your partners, select **Reserved for partner**.
5. Repeat steps 3 and 4 for each territory in your sales team's organizational structure.

Adding operators

Create operator records for all users so that you can define the permissions for each user. For more information about operators, see [Sales team](#).

A Global Sales Ops operator can add operators and grant read, update, and create permissions to work objects belonging to any territory. A Local Sales Ops operator can add operators and grant read, update, and create permissions to work objects belonging to their primary territory and any children under that territory.

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
2. In the navigation pane, click **Operators**.
3. Click **Create operator**.
4. On the **Operator Details** form, complete the required fields and enter optional information as needed for your implementation:
 - a. **Optional:**
Optional: Enter a value in the **Partner** field if you are setting up an operator for a partner.
 - b. Click **Password** to enter a password for the operator and then click **Submit**.
 - c. Click **Next**.
5. On the **Operator Access** form, accept or clear the **Use default operator access** option and then click **Next**.
 - If you accept this option, select an **Operator Type** and a **Default Territory**. The new operator derives access and permission rights from the default territory.
 - If you clear this option, select an **Operator Type** and a **Primary Territory**. You can then define the new operator's access and permission rights for the primary territory. You can also assign the operator to additional territories and define access and permission rights for each additional territory assigned.
6. **Optional:** On the **Sales Goals** form, enter sales goals for the operator and click **Next**.
If you enter sales goals, then real-time dashboards and charts provide visibility into sales performance. You can add goals during operator creation, or a Sales Operations operator can add goals after the operator is created.
7. **Optional:** Enter HR details and click **Finish**.

Adding partners

Configure the external agencies with whom you work. For more information about partners, see [Sales team](#).

A Global Sales Ops operator can set up and administer each partner and delegate user (operator) administration to a Local Sales Ops operator at the partner organization.

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
2. In the navigation pane, click **Partners**.
3. Click **Create Partner**.
4. Complete the required fields and enter optional information as needed for your implementation.
5. **Optional:** To give the partner access to the application and assign the partner to a territory, select **Operator Access Required**.
6. Click **Create**.

Configuring prospect and customer data

The Pega Sales Automation application is the system of record for the prospects and potentially the customer data that is used by the sales team, including the organization, accounts, households, and contacts.

- [Prospect and customer data types](#)
- [Importing prospect and customer data](#)

Prospect and customer data types

The types of prospect and customer data that are required for your implementation depend on the selling model that you select. Prospect and customer data can include information about the following entities:

Organization

An organization is a specific business, holding company, corporation, or association. The organization represents the top level of a business hierarchy that contains one or more child entities in the form of accounts. Organizations are used only in the business-to-business (B2B) selling mode.

Accounts

An account is an economic decision making unit that represents the logical or physical group to which products or services are sold. Accounts are used in both the business-to-business (B2B) and business-to-consumer (B2C) selling modes.

Contacts

A contact (person) is a prospect or a customer.

- In the business-to-business (B2B) selling mode, a contact is usually an employee of or somehow affiliated with the organization to which you are selling.
- In the business-to-consumer (B2C) selling mode, a contact is the person interested in making a purchase.

You can associate a contact with an account, organization, or another contact to facilitate cross selling and communication. The following relationship type codes are available for defining contact relationships in the application:

- C2A - Indicates a contact-to-account relationship.
- C2O - Indicates a contact-to- organization relationship.
- C2C - Indicates a contact-to-contact relationship.

You can further define the nature of these relationship types. For example, a C2C relationship can be a relationship between colleagues in an organization or between family members in a household.

Households

A household is a grouping of contacts in which one of the contacts is designated the head of household. A household usually represents members of a family, or contacts living at the same address. A household can also be used to group any set of contacts that share a commonality and have a primary contact person. Households are used only in the business-to-consumer (B2C) selling mode.



Note: The household data type is optional. If you plan to use household data, you must activate the data type for your implementation. For more information, see [Activating application features](#).

Importing prospect and customer data

Use the data import tools to import your existing prospect and customer data into Pega Sales Automation.

1. Identify the types of prospect and customer data to import, based on your selling model.

Selling model	Import these data types
Business-to-business (B2B)	• Organizations • Accounts • Contacts
Business-to-consumer (B2C)	• Accounts • Contacts • Households
Business-to-business and business-to-consumer (B2B_B2C)	• Organizations • Accounts • Contacts • Households

2. Import the data using one of the methods described in [Pega Sales Automation data import](#).

Configuring data loaders extensions

In Pega Sales Automation 8.4, the dependency with flow execution while uploading data by using the File Listener and Data Import wizard for organizations, accounts, and contacts has been removed. The utilities flow logic has been redesigned to enable quick import.



Note: If you have overridden flows during your application implementation, it is recommended that you use utilities. Up to the completion of implementation, the `isQuickImportEnabled` when conditions should be overridden with a false condition in implementation classes.

By default, the `isQuickImportEnabled` condition for organizations, accounts, and contacts is set to true. For other work objects, it is set to false. To configure data loaders extensions, perform one of the following actions:

- To use utilities, set `isQuickImportEnabled` to true.

- To use flows, set `isQuickImportEnabled` to false.

For more information, see [Pega Sales Automation data import](#).

Opportunity splits between sales team members

As a sales representative or a sales manager, you can now split opportunities with team members to improve collaboration and share the revenue.

Opportunity owners and sales managers can customize the default split values, as well as add or remove team members. To enable opportunity splits, in the navigation panel of App Studio, click **Settings** >

Application Settings, and then select the **Opportunity Splits** check box. After you enable opportunity splits, on the opportunity page, the system shows the default split values.

After you enable opportunity splits:

- On the opportunity page, the system shows the default split values, which you can customize.
- Opportunity splits are listed as additional columns to some dashboard widgets, for example, pipeline by territory, and quota attainment. For example: Terry has 50% allocation on the 'Xyz' opportunity with \$10,000 revenue. Terry's pipeline includes this allocation: \$5000 and if the opportunity is closed as won, \$5000 is added to his sales quota.
- On the Forecast page, sales managers can filter data by territory and sales representative to view and analyse their performance with split values for each category. Use forecasting with revenue splits to predict revenue splits, based on unweighted, predictive, and weighted opportunity amounts. To enable the split view for forecasting, click **Advanced** and turn on the **Show revenue splits** switch.
- [Configuring opportunity splits](#)

Configuring opportunity splits

Enable the opportunity splits feature so that multiple team members can collaborate on an opportunity, and share the opportunity revenue.

1. In the navigation pane of App Studio, click **Settings** > **Application Settings**.
2. On the **Features** tab, select the **Opportunity Splits** check box.
3. In the navigation panel of **App Studio**, click **Reports**.
 - a. On the **Reports** screen, search for and open an opportunity report.
 - b. Click **Actions** > **Export** and export your opportunity data as a CSV file.
4. In the file that you export, update the data to include the team members and their split allocations. To avoid mapping errors, ensure that the total percentage values for a team equal exactly 100%. For more information about how to enter data correctly, see [Pega Sales Automation entities](#).
5. In the navigation panel of Dev Studio, click **Records** > **SysAdmin** > **Dynamic System Setting**. For the initial split data allocation, import only Opportunity ID's in the CSV file by using the **Add or Update** option. By doing this the opportunity owner gets 100% and the rest 0%. For the optional process performance, set the batch size value to 1.
 - a. Search for and open the `import/BatchSize` dynamic system setting.
 - b. Set the value to 1.
 - c. Click **Save**.
6. Import the updated data by using the **Add or update** option in the data import wizard. For more information, see [Pega Sales Automation data import](#).

The opportunity splits values are updated for each sales team.

Enabling application features

Pega Sales Automation includes a set of core and optional features that you can implement to support your sales team.

For information and instructions on how to implement the features required for your implementation, see the following topics:

- [Activating application features](#)
- [Implementing the Sales methodology](#)
- [Implementing case types](#)
- [Implementing products and competitors](#)
- [Implementing forecast management requirements](#)
- [Implementing activity and task management requirements](#)
- [Implementing sales goals](#)
- [Implementing reporting](#)
- [Implementing a mobile app](#)
- [Implementing Marketing features](#)
- [Implementing engagement maps](#)
- [Implementing organization trend data](#)
- [Implementing Pega Co-Browse](#)
- [Configuring tagging for work objects and Pulse posts](#)
- [Configuring Pulse](#)
- [Implementing Pega Call](#)
- [Implementing Pega Knowledge](#)
- [Implementing artificial intelligence](#)
- [Implementing third-party integrations](#)
- [Implementing the security model](#)

Activating application features

Use App Studio to activate optional features in Pega Sales Automation. After you activate features in the application, you may need to complete additional set up steps to configure the features for your implementation.

1. In the navigation pane of App Studio, click **Settings** > **Application settings**.
2. On the **Features** tab, verify the **Selling mode** that you specified when you created your application. You can change the selling mode by clicking one of the available selling mode options.
3. Review the following list of available features and activate features that match your business requirements:

- To activate artificial intelligence insights, select the **Artificial intelligence insights- opportunity insights, lead ranking, and sales coach** check box.

When you select this check box, you can see all AI capabilities and enable each one separately by selecting individual check boxes.

By default, all options are selected. To disable an option, clear the check box.

For more information about artificial intelligence capabilities in Pega Sales Automation, see [Artificial intelligence features in Pega Sales Automation](#).

- To activate the engagement maps feature, click the **Engagement maps** check box.
For more information, see [Engagement maps requirements](#).
- To activate integration with Pega Knowledge, click the **Knowledge management** check box.
For more information, see [Pega Knowledge integration](#).
- If your organization works with external agencies, resellers, and sales partners to sell your products, click the **Partner relationship management (PRM)** check box to activate partner management functionality.
For more information, see [Partners](#) and [Adding Partners](#).
- To activate the household data type, click **Households** check box.
For more information, see [Configuring prospect and customer data](#).
- To activate integration with Pega Call, click the **Computer telephony integration** check box.
For more information, see [Defining computer telephony integration \(CTI\) requirements with Pega Call](#).
- If you are using the forecast management feature, click the **Close plans** check box to activate qualitative forecasting.
For more information, see [Forecast management requirements](#).
- To activate integration with Workforce Intelligence, click the **Workforce Intelligence** check box.
For more information, see [Workforce Intelligence integration](#).
- To activate geolocation services using Google Maps, click the **Google Maps & address geo-coding - separate license required from Google** check box and then enter your Google Maps API key in the **Google key** field.
For more information, see [Using the Address Map control](#).
- To activate the pre-integrated Pega Customer Relationship Management features, select **Enable CRM mode**.
- To activate Dun and Bradstreet Direct 2.0 Data so that you can stream and integrate D&B Data using the D&B Direct API, select **Enable D&B**.
For more information, see [Hoover's Dun & Bradstreet Direct 2.0 Data](#) and [Integrating Pega Sales Automation with Hoover's Dun & Bradstreet Direct 2.0 Data connector](#).

Implementing the Sales methodology

In Pega Sales Automation, you use Pega's case life cycle management tools to model your sales methodology and guide sales representatives through each stage of the sales process.

The following topics provide information about how to implement the sales methodology:

- [Defining the sales methodology](#)

- [Configuring the sales methodology](#)

Defining the sales methodology

Pega Sales Automation includes several predefined case types that represent the processes and work objects that drive your organization's sales life cycle. Select the case types to include in your application when you build your implementation layer. If you need additional case types not provided in the application, create them during your implementation.

For information about case life cycle management, see [Defining the case life cycle](#).

1. Review the sales processes for your organization.
 - a. Define requirements for managing leads.
 - b. Define requirements for managing opportunities.
2. Map the lead and opportunity requirements to stages and steps in the lead and opportunity case types.
3. Determine if you need to import existing lead and opportunity data.
4. Determine the type and source of the existing lead and opportunity data that you need to import.

- [Lead and opportunity case types](#)

Lead and opportunity case types

The two main case types that you use to implement your sales methodology are the lead and opportunity case types.

Lead

A lead is a prospect (customer or non-customer) that your sales team evaluates to determine if they are qualified to advance to the probability of a sale. When you qualify a lead, it is converted to an opportunity.

When you create the implementation layer, the lead case type contains no stages. You can customize the lead case type by adding stages to reflect the processes your sales team follows to qualify a lead. You can further divide your lead case type stages into steps that define the tasks or activities that your sales representatives perform within each stage, such as managing campaign responses, creating lead profiles, or scheduling appointments.

Opportunity

An opportunity is a qualified deal, or a potential sale that you track, nurture, and manage. Opportunities are the foundation of your sales pipeline, goal attainment, and sales forecast.

When you create the implementation layer, the opportunity case types (B2B Sales Opportunity and B2C Sales Opportunity) each contain a set of pre-configured stages and steps.

You can customize the opportunity case types by modifying the Pega-provided stages or adding your own stages and steps to reflect the processes your sales team follows to move an opportunity through the sales pipeline. You can also modify the Pega-provided steps or add your own steps to define the tasks or activities that your sales representatives perform within each stage, such as identifying competitors, developing proposals, and scheduling appointments.

Configuring the sales methodology

Use the lead and opportunity case types in Pega Sales Automation to implement your organization's sales methodology.

The **Lead** stage comes with following pre-configured set of stages: Assigned, Marketing Qualified, Sales Qualified, and Closed.

What to do next:

After you configure the opportunity case type, it is used to guide your sales representatives through every step of the sales process for each opportunity you create in the application.

Implementing case types

Modify the application-provided case types to meet your business needs.

Modifying an opportunity case type

You can modify an opportunity case type to reflect the stages and steps in your sales methodology.

The Pega-provided sales methodology for Pega Sales Automation does not force sales representatives to follow processes in a fixed sequence. As a result, all the processes within the different stages of an opportunity are modeled as optional processes. You can update the case type rule for your implementation to use optional processes or a combination of automatically launched and optionally launched processes within each stage.

The stage a case is on and the recommended processes for each stage are shown in the chevron view in the Review harness. Each of the recommended processes within a stage is shown as a link.

- For automatically launched processes, clicking a stage link launches the associated assignment
- For optional processes, clicking a stage link launches the processes for the user to perform the recommended action.

The section `StageChevronView` renders the overall case life cycle and includes a section `crmDisplayStages` that displays the stages and processes within a stage. This section is parameterized to enable or disable links for optional processes from completed stages.

Mandatory processes are modeled as automatically launched processes. When you try to move the case from one stage to another, a validation rule checks if all mandatory processes in the current stage are completed.

- B2B Sales Opportunity - To add a stage to a B2B Sales Opportunity case type, modify the PegaCRM - Work-SFA-Opportunity-Biz.pyDefault case type rule.
- B2C Sales Opportunity - To add a stage to a B2C Sales Opportunity case type, modify the PegaCRM - Work-SFA-Opportunity-Ind.pyDefault case type rule.

Complete the following steps to add stages and steps to a B2B Sales Opportunity or B2C Sales Opportunity case type:

Adding stages to an opportunity case type

A stage represents a unique phase in your sales methodology, for example, the qualification or proposal phase.

1. In the navigation pane of Dev Studio, click **Case types**.
2. Right-click a case type that you want to add stages to and click **Open**.

3. Open the **Stages** tab.
4. In the **Primary Stages** section, complete one of the following steps:
 - To insert a stage after the last stage in the list, click **add primary stage**.
 - To insert a stage after an existing stage in the list, click the existing stage name and click **add primary stage**.

The order of the stages in the **Primary Stages** section determines the order in which the stages display in the application. You can drag and drop stages in the **Primary Stages** section to change their order.

5. Click **Save as**.

Specifying a field value for a new opportunity stage

Specify the field value for a new opportunity stage to support the localization of the stage name in portal displays, reports, and user forms.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand **Data Model** and right-click **Field Value**.
3. Click **Create**.
4. Complete the fields on the Create Field Value form:
 - In the **Label** field, enter the stage name.
 - In the **Field Name** field, enter `OpportunityStage`.
 - In the **Apply to** field, enter the implementation class.
5. Click **Create and Open**.
6. Verify that the **To** field displays the name of the stage as it should appear in the user interface.
7. Click **Save**.

Adding steps to an opportunity stage

A step is a unit of work in a stage, for example, the development or delivery of a proposal in the proposal stage.

1. In the navigation pane of Dev Studio, click **Case types**.
2. Right-click a case type that you want to add steps to and click **Open**.
3. Open the **Stages** tab.
4. In the **Primary Stages** section, click the stage for which you want to add a step.
5. In the **Optional Processes** section, click **Add process** and enter the name of the process that you want to launch when the step is selected.

You can select an existing process flow or create a new one. If you create a new process flow, you must save it before you save the case type rule.
6. Click **Save as**.

Implementing products and competitors

Product and competitor information helps your sales team to align opportunities with product offerings and to differentiate your products when they make a sales pitch.

Your products are the plans that you sell to your customers. Your competitors are other organizations that sell similar products to the same base of customers.

- [Defining products and competitors](#)
- [Setting up products](#)
- [Setting up competitors](#)

Defining products and competitors

Your products are the goods and services that you sell to your customers. Your competitors are other organizations that sell similar products to the same base of customers. Enter product and competitor information in the application to align opportunities with product offerings and to identify your competitors so that you can differentiate your products when you make a sales pitch. Pega Sales Automation supports Categories, Families, Groups, and Products hierarchy.

Before you begin: Before you enter product information in the application, define your product hierarchy.

1. Define product categories, for example Life Insurance and Health Insurance.
2. Define product families, for example Individual Life and Individual Health.
3. Define product groups, for example Term Life and Supplemental Health.
4. Define products, for example 5-year Renewable Term and Individual Accident.
5. Define the selling mode for your products: Business (B2B), Individual (B2C), or All.
6. Define competitor and data import requirement.
 - a. Identify competitors who are selling similar products.
 - b. Determine if you need to import existing product and competitor data.
 - c. Determine the type and source of the existing product and competitor data that you need to import.

What to do next:

If your products are maintained in an external system of record, determine how you will connect to the external system. For more information, see [Setting up products](#).

Setting up products

Products are the goods and services that you sell to your customers. A Sales Ops operator sets up and manages your products in the Pega Sales Automation application. You can set up your products by importing existing product data, or by entering the product data manually.

To set up products manually, complete the following steps:

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
 2. In the navigation pane, click **Tools**.
 3. Click **Products > Create product**.
 4. Complete the necessary fields and click **Submit**.
 5. Repeat steps 3 and 4 for each level of your product hierarchy.
- [Configuring an external product data source](#)

Configuring an external product data source

If your products are maintained in an external system of record, you must update the `crmAppExtModel` data transform and the `D_InterfaceProductListFromExternal` data page for your implementation.

1. In the header of Dev Studio, search for and select the `crmAppExtModel` data transform.
2. Locate the `.IsProductSORExternal` **Target** and enter `true` in the **Source** field for the target.
3. Click **Save**.
4. In the Dev Studio header search text field, search for and select the `D_InterfaceProductListFromExternal` data page.
5. In the **Data sources** section, configure the external system of record for your products.
Review the `crmBuildSampleInterfaceData` and `ConstructSampleProductInstance` data transforms to see a sample interface data model. For more information, see [Data page rules - Using the Definition tab](#).
6. Click **Save**.

Setting up competitors

Competitors are other organizations that are selling products similar to yours. A Sales Ops operator sets up and manages your competitors in the Pega Sales Automation application.

You can set up your competitors by importing existing competitor data, or by entering the competitor data manually.

To set up competitors manually, complete the following steps:

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
2. In the navigation pane, click **Tools**.
3. Click **Competitors > Create competitors**.
4. Complete the fields on the Create competitor form.
5. Click **Submit**.

Implementing forecast management requirements

Forecast management provides a summary of your sales pipeline for the year, broken down by fiscal quarter and forecast category. You can use forecast management to review pipeline status for an entire territory or for an individual operator.

The basic features of forecast management are ready for use with Pega Sales Automation and do not require additional configuration during implementation. However, if you want to use the close plan feature with forecast management, you must activate it for your implementation. For more information, see [Activating application features](#).

Implementing activity and task management requirements

An activity is any meaningful communication between a sales representative and a contact that is designed to drive a sale or develop a professional relationship. Activities have a purpose, an outcome, and an interaction date, and can have follow-up tasks. Activities are used in both the business-to-business (B2B) and business-to-consumer (B2C) selling models.

To manage activities and tasks in Pega Sales Automation, you must define your activity data model and import requirements.

1. Determine if you need to import existing activity and task data.
2. Determine the type and source of the activity and task data that you need to import.

Implementing sales goals

Sales goals give sales managers the ability to plan quarterly or monthly goals for sales representative based on a variety of metrics. Goals, by default, are based on revenue but can be driven by other metrics such as activities, number of opportunities closed, or number of leads created. You can track goal achievement by using the Quota Attainment widget on the dashboard. You can view sales goals for your operators in the Sales Ops portal.

- [Defining requirements for sales goals management](#)
- [Setting up time periods for sales goals management](#)

Defining requirements for sales goals management

Sales Ops operators can enter sales goals for sales representatives when setting up operator IDs or when updating the Sales Goals tab in existing operator records.

1. Determine the start and end date of the fiscal year for your sales organization.
2. Determine the sales goals reporting periods for your sales organization.
3. Determine if you need to import existing sales goals data.
4. Determine the type and source of the existing sales goals data that you need to import.

Setting up time periods for sales goals management

Time periods are required if you plan to create and monitor sales goals. Pega Sales Automation uses the time periods you set up to track progress toward meeting sales goals.

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
2. In the navigation pane, click **Tools**.
3. Click **Time periods > Create time period**.
4. Complete the fields on the Time period form.



Note: Create the fiscal year time period first and then create the quarters as children to the fiscal year. If you measure by month, you can add months as children to each quarter.

5. Click **Submit**.

Implementing reporting

Based on your login and portal, Pega Sales Automation provides numerous pre-configured reports to monitor sales activity and to monitor workforce productivity. These standard reports are driven by the case lifecycle and work object infrastructure, as well as rules for goals and deadlines used in managing leads and opportunities.

- [Defining reporting requirements](#)
- [Reports](#)

Defining reporting requirements

Consider your reporting requirements early in the planning process.

For more information about the reporting capabilities in Pega Sales Automation, see [Reporting](#).

1. Review the description of each application-provided report, identify the expected volume of data, and determine how often you expect to run each report.
2. Identify any required metrics that are not already provided by the application.

Reviewing existing reports

The application includes numerous reports. Identify the application-provided reports that meet your business needs.



Note: Pega Sales Automation includes the standard Pega Sales Automation reports to help you monitor and analyze your sales activity. For descriptions of each report, see [Pega Sales Automation reports](#).

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
2. In the navigation pane, click **Reports**.
3. Review each of the reports in the **Public categories** section.
4. Determine which operator roles need the report, what it contains, and when and why it's needed.

Reports

In addition to the library of Pega Platform reports, Pega Sales Automation provides several pre-configured reports that are customized for the various work objects, cases, and business processes used in the application. These reports provide you with real-time insights into case inventories and aging so that you can respond with the agility that is needed to comply with your internal and external service level agreements as well as compliance mandates.

In addition to the library of Pega Platform reports, Pega Sales Automation provides several pre-configured reports that are customized for the various work objects, cases, and business processes used in the application. These reports provide you with real-time insights into your sales activities so that you can respond with the agility that is needed to manage the sales pipeline and meet your sales goals.

To create other reports, see [Reporting](#).

Implementing a mobile app

The Pega Infinity Mobile Client is a service that enables the quick building of mobile apps from the Pega Platform application. You need a license for the Pega Infinity Mobile Client. For information about Hybrid Client mobile app, see [Pega Sales Automation Hybrid Client mobile app deployment](#).

- [Completing prerequisite tasks before building the Pega Sales Automation mobile app](#)
- [Building the Pega Sales Automation mobile app](#)

Completing prerequisite tasks before building the Pega Sales Automation mobile app

Before you build the Pega Sales Automation mobile app, review the following items:

These steps apply to both Pega Infinity Mobile Client and Hybrid Client mobile apps.

1. [Verify that Pega Platform settings are complete](#).

These settings are configured during the system configuration phase by the Pega Platform administrator in your organization. If they are not configured, contact the administrator to verify that your organization has been licensed to use the Pega Mobile Client app building feature. If you do not have a license, contact your account executive.

2. Determine if the organization needs to customize the mobile app.

For more information about the Pega Mobile Client API, see [Pega Mobile Client API reference](#) and [Searching device contacts with Pega Mobile Client API](#).

If you are planning to build a mobile app, consider configuring your mobile app's functionality with support of device-specific features, such as a barcode scanner.

3. Ensure that you have a certificate set for the Android or iOS mobile platform. A certificate set defines various settings which identify the Android or iOS application that you are creating.

For more information about the certificate set for the Android platform, see [Creating an Android certification set](#).

You need to generate the iOS certificate set in the Apple Developer Portal ahead of time. For more information, see [Generating mobile certificates for the iOS platform](#).

4. Determine if the organization wants to brand the mobile app.

You can customize the branding of your mobile app to serve the organization's specific requirements and needs. You can upload your own images for the splash screen and menu icon to the Pega Platform during the app customization process. If you do not provide custom assets, a default Pega Platform branding is applied. For more information, see [Preparing custom branding assets](#).

Building the Pega Sales Automation mobile app

After you determine the organization's needs in the [Completing prerequisite tasks before building the Pega Sales Automation mobile app](#) section, you can build your Pega Infinity Mobile Client mobile app.

1. [Review app features in Pega Sales Automation sample application](#).
2. [Review Pega Sales Automation rules and patterns for reuse in the mobile app](#).
3. Use the Pega Platform-provided functionality to enable office access for your mobile app. For more information, see [Enable offline access](#).
4. [Build and brand your mobile app](#).
5. [Deploy your mobile app](#).



Note: For more information, see [Pega Mobile](#).

6. **Optional:** [Set up optical character recognition \(OCR\) scanning](#).
For information about the Hybrid Client mobile app deployment, see [Pega Sales Automation Hybrid Client mobile app deployment](#).
- [Reviewing app features in the sample application](#)
 - [Reviewing rules and patterns for reuse in the mobile app](#)
 - [Deploying your mobile app](#)
 - [Setting up optical character recognition \(OCR\) scanning](#)

Reviewing app features in the sample application

Before you deploy the Pega Sales Automation mobile app, install the Pega Sales Automation sample application to explore the mobile app features. For more information, see [Working with Pega Sales Automation sample data](#).

Reviewing rules and patterns for reuse in the mobile app

You can reuse or extend various Pega Sales Automation rules and patterns to control the appearance of your Pega Infinity Mobile Client mobile app based on your business requirements.

- To add or remove navigation items, in Dev Studio, search for and override the *SAPortalMenuItems_Mobile* rule.
- To customize the appearance of the app with cascading style sheets, override the *sa-style-helper-pimc-ext* file in your implementation layer by adding your custom code.
- To use the Late for meeting notification, override and enable the *GenerateAppointmentNotifications* job scheduler.



Note: To override and enable the *GenerateAppointmentNotifications* job scheduler, in Admin Studio, click **Resources** > **Jobs**, search for *GenerateAppointmentNotifications*, and then click **Override**.

- To use the Today's meeting and Appointments notifications, configure Microsoft Exchange services.



Note: For basic authentication, update the *EWS_Integration_AuthProfile* authentication profile, or *EWSOAuthAuthenticationProfile* for OAuth2 authentication. In addition, users must have email accounts of the same domain as added in the used authentication profile. Credentials added in the authentication profile must have the impersonation permissions added. For more information, see [Microsoft Exchange and Microsoft Outlook integration in Pega Sales Automation](#).

Deploying your mobile app

To create a Pega Infinity Mobile Client mobile app channel or an SDK app for the Android or iOS platforms, perform the following steps:

1. In the navigation pane of App Studio, click **Channels**, and open the out-of-the-box Pega Sales Automation Mobile Client.
2. In the top-right corner of the screen, click **Build app**.
The system automatically creates a portal with the same name as the mobile app name.
3. Switch to Dev Studio.
4. In the header of Dev Studio, open your current access group to associate this access group with a portal, and configure which workspaces or web channels are available to the users working in your application:
 - a. In the bottom-left corner, click the **Profile menu** button > **Access group**.
 - b. On the **Definition** tab, in the **Available portals** section, click **+ Add portal**.
 - c. In the **Name** field, press the Down Arrow key and select the portal name.
 - d. To make this portal the default for all operators in the access group, click the radio button next to it.
5. Click **Save**.

For information about the Hybrid Client mobile app deployment, see [Pega Sales Automation Hybrid Client mobile app deployment](#).

Setting up optical character recognition (OCR) scanning

With the Pega Sales Automation mobile app you can scan a business card to create a contact in the application.

After you set up the OCR integration, you can click **Scan card** on a contact form in the app to use your mobile device's camera to take a picture of a business card. The app receives the business card image and sends it to a third-party OCR software vendor, where it is parsed into a standard output, such as XML or JSON. The result is then mapped back to Pega Sales Automation to create a new contact record.

Follow these steps to set up the OCR integration for your implementation:

1. In Dev Studio, complete one of the following steps:

- Identify an existing ruleset and ruleset version for the OCR integration.
- Create a ruleset and ruleset version for the OCR integration and then add the ruleset and ruleset version to the application stack.

For information about creating and saving rulesets, see [Creating a ruleset and ruleset version](#).

2. Create a data model in the selected ruleset that reflects the data package result from the third-party software used in the `PegaCRM-Int-BusinessCard-` class:

- If the third-party software uses an XML object, create a data structure to hold the XML structure.
- If the third-party software uses a JSON object, create a data structure to hold the JSON object.

For more information about creating data models, see [Adding a field group to your data model](#).

3. Save the following extension point rules to the OCR ruleset:

- The `GetCardData` activity, which parses the business card data for the OCR vendor.
- The `MapBusinessCardDetails` data transform, which maps the data received from the OCR vendor to the contact in Pega Sales Automation.

4. Open the `ScanBusinessCard` activity and save it to the OCR ruleset:

- Click the **Pages & Classes** tab.
- In the **Class** field for `BusinessCardParsedData`, enter the newly created class name.
- Click **Save as**.

Implementing Marketing features

Pega Sales Automation includes numerous Marketing features to support the sales process, including: local campaigns, next-best-offer strategies, and next-best-action strategies.

- [Local campaigns](#)
- [Next-best-offer recommendations](#)
- [Enabling local campaigns and next-best-offer recommendations](#)
- [Next-best-action strategies](#)

Local campaigns

As a sales representative or a sales manager, you can use the local campaigns feature to create and send personalized email and SMS campaigns to contacts in the Pega Sales Automation application. When a contact accepts a campaign offer, a lead is automatically generated in the application.

Local campaigns are supported for both business-to-business (B2B) and business-to-consumer (B2C) selling models.

To implement local campaigns in Pega Sales Automation, you must first integrate Pega Sales Automation with Pega Marketing. You can then activate the feature in Pega Sales Automation.

- For information about integrating Pega Sales Automation with Pega Marketing, see [Configuring your applications for Pega Customer Relationship Management](#).
- For information about activating local campaigns in Pega Sales Automation, see [Enabling local campaigns and next-best-offer recommendations](#).

Next-best-offer recommendations

Pega Sales Automation uses Pega Customer Decision Hub to identify the top sales offer across a sales representative's book of business. Adaptive models guide sales representatives and sales managers to the offers that have the highest propensity to close. When a prospect accepts an offer, an opportunity is automatically generated.

To implement next-best-offer recommendations in Pega Sales Automation, you must first integrate Pega Sales Automation with Pega Marketing. You can then activate the feature in Pega Sales Automation.

- For information about integrating Pega Sales Automation with Pega Marketing, see [Configuring your applications for Pega Customer Relationship Management](#).
- For information about activating next-best-offer recommendations in Pega Sales Automation, see [Enabling local campaigns and next-best-offer recommendations](#).

Enabling local campaigns and next-best-offer recommendations

To implement local campaigns and next-best-offer recommendations in Pega Sales Automation, you must first integrate Pega Sales Automation with Pega Marketing. You can then activate the features in Pega Sales Automation.

1. In the navigation pane of App Studio, click **Settings**.
2. Click the **Pega Marketing** tab.
3. To activate local campaigns, click the **Local campaigns** check box.
4. To activate next best offers, click the **Next best offers** check box.
5. In the **Marketing URL** field, enter the URL for your Pega Marketing implementation, for example, `http://NN.NN.N.NNN:8080/prweb/PRSOAPServlet`.
6. In the **Customer lifetime value (CLV)** fields, enter the maximum, average, and minimum CLV values for your organization.

Customer lifetime value (CLV) is a prediction of the net profit that is generated as a result of an organization's entire future relationship with a customer. Pega Marketing uses CLV values to identify customers with the potential to grow, which allows you to more effectively target your marketing efforts. Customer lifetime value is applicable to both business-to-business (B2B) and business-to-consumer (B2C) customers.
7. Click **Save**.

Next-best-action strategies

Pega Sales Automation uses Pega Customer Decision Hub and Decision Management capabilities to identify the next best action for every opportunity in the application, rank the top actions, and display the results in the Next best actions widget on the sales representative's dashboard.

Next-best-action strategies use predictors such as digital activity, service activity, products sold, and sales representative activity to determine the next best action for every opportunity.

- For more information, see [Next best action in Pega Sales Automation](#).

Implementing engagement maps

Engagement maps are account and territory planning tools that provide a visual map of your product penetration levels. Engagement maps can help you to identify territories and accounts where your product penetration level is low.

Before you begin:

Before you can view engagement maps, you must set up a product hierarchy in Pega Sales Automation. For more information, see [Defining products and competitors](#).



Note: The optional engagement maps feature is disabled by default and therefore not visible in the user portals. To use this feature, you must activate it in your application. For more information, see [Activating application features](#).

Implementing organization trend data

Organization trends provide visibility into your interactions with customers across multiple channels, including web activity, email traffic, interactions with contacts, and contact growth. You can source data for your trend reports from Pega Sales Automation tables or from external databases or data warehouses. Organization trend data is displayed in the corresponding organization record in Pega Sales Automation.

- [Defining organization trend requirements](#)
- [Mapping data for organization trends](#)

Defining organization trend requirements

If you plan to track organization trends, you must map the application to the source of the trend data.

1. Determine the source of trend data.
2. Map the customer engagement data class to the table containing the desired data. For more information, see [Mapping data for organization trends](#).

Mapping data for organization trends

Organization trends provide visibility into your interactions with customers across multiple channels, including web activity, email traffic, and interactions with contacts.

Before you begin:

You can source data for your trend reports from Pega Sales Automation tables or from external databases or data warehouses.

You must map the customer engagement data class in the application to the table containing the desired data.

1. In the Dev Studio header search text field, search for and select the `PegaCRM-Int-CED-Aggregated` data class.

2. Map to the desired database table:

- For an external table, map to the external database or data warehouse table.
- For an internal table, map to the `Pega Sales Automation CED_SAMPLE_DATA` sample table.

3. Click **Save**.

For more information about completing the fields on the `PegaCRM-Int-CED-Aggregated` data class form, see [Database Class Mappings tab](#).

Implementing Pega Co-Browse

With Pega Co-Browse, your customers and prospects can collaborate with sales representatives by sharing a web browser session on their PC, tablet, or mobile phone .

- [Defining Pega Co-Browse requirements](#)
- [Configuring Pega Co-Browse](#)

Defining Pega Co-Browse requirements

Define how your organization will implement Pega Co-Browse.

1. Determine which web pages that you would like Pega Co-Browse to work on.
2. Determine which website fields require masking.
3. Determine whether the Pega Co-Browse window must be customized to match branding standards at your site.
4. Define a message that you want to appear before the viewer connects with the presenter.
5. Determine whether to enable remote control. Remote control allows the viewer to take control of the presenter's screen.

Configuring Pega Co-Browse

You can activate co-browse functionality for your implementation by logging in to the Pega Co-Browse website (authorization required). The JavaScript on the site's **Install** tab must be included in your website to allow co-browsing. For more information about how to activate co-browse functionality, see the [Pega Co-Browse Implementation Guide](#) on the [Pega Co-Browse product page](#).

You must include the following rules in your implementation application to use Pega Co-Browse:

Rule name	Description
<code>crmSessionCode</code>	This property is used to store the URL for websites that have been enabled for co-browse. The user can be sent to different parts of the website, based on need.
<code>crmWorkMenu</code>	This navigation rule is used to allow the user to enter the session code or to initiate a new co-browse session.

Rule name	Description
crmCoBrowseSessionCode	This flow captures the session code and displays the co-browse screens. It also creates an activity on completion to track the session.
crmCoBrowseEmailLink	This flow captures the customer email and the destination URL for the co-browse enabled website. It also creates an activity on completion to track the session.
crmCoBrowseEmailLink	This rule captures the customers email and the destination URL.
crmCoBrowseEmail	This rule contains the body of the email being sent to the customer.
crmCoBrowseSessionCode	This rule captures the session code from the customer.
crmCoBrowseAttributes	This rule sets up values for creating an activity on the conclusion of the co-browse session.
crmCoBrowsePost	For an Email Link co-browse session, this rule initiates the <code>crmCoBrowseTypeSetValues</code> data transform. For an Enter Session co-browse session, this rule builds the destination URL.
crmCoBrowseTypeSetValues	This rule sets up the Email and Session code for the Email Link session type and clears all values for Enter Session Code sessions.
crmStartCoBrowse	This control holds the code to display the co-browse session.

For information about how to configure the rules required for Pega Co-Browse integration, see [Rules development](#).

Configuring tagging for work objects and Pulse posts

Every Pega Sales Automation user can tag work objects and use these tags as sorting phrases to organize data in a personalized way.

You can add a tag for work objects (organizations, accounts, contacts, leads, households, appointments,) or Pulse posts.

To add a tag, open either a work object or a Pulse post and click **Actions > Add tag**. To view all objects that have an associated tag, in the Pega Sales Automation navigation pane, click **Tags**.

To tag other objects than mentioned above, configure tags the following way:

1. In the Dev Studio header, search for `crmworkmenu` and select a rule of the object that you want to configure.
2. On the **Editor** tab, double-click the **Tags** line to open the tags configuration section.
3. On the **Actions** tab, in the **Local Action** field, enter `pzAssociateTags`.
4. Click **OK** and **Save**.

What to do next: For more information, see [Tagging a new message in Pulse](#).

Configuring Pulse

Pulse adds social activity stream capabilities to your user interface. With Pulse, application users can collaborate in real time, share comments, files, and URLs with other users, and take action directly from an activity stream.

Pulse offers contextual behavior so that its scope depends upon where you use it in the Pega Sales Automation application. You can broadcast to an entire sales team or share information only with users who have access to a specific lead or opportunity. For more information about the Pulse feature, see [Communicating with users by using Pulse](#).

To configure Pulse, see [Configuring Pulse email notifications](#).

Implementing Pega Call

Pega Call provides robust computer telephony integration (CTI) support for Pega Sales Automation users, including click-to-dial functionality for outgoing calls and adaptive screen pops for incoming calls.

- [Defining computer telephony integration \(CTI\) requirements with Pega Call](#)
- [Configuring computer telephony integration using Pega Call](#)

Defining computer telephony integration (CTI) requirements with Pega Call

To plan the connectivity to your telephony infrastructure, complete the following tasks:

Before you begin:

After you install and configure Pega Call, you must activate CTI functionality for the application and then add the `PegaCTI:CSR` access role to all the access groups for which you are providing CTI access. For more information, see [Activating application features](#).

1. Understand which telephony vendor that your site uses.
2. Review the [Pega Call CTI Implementation materials](#) to understand the configuration for your vendor.
3. Meet with the technical resources responsible for configuration and maintenance of the Contact Center Telephony and CTI infrastructure at your site to get an understanding of the current configuration and desired call flows and customer experience. Ensure that a resource from the telephony team is available to work with the implementation team.

4. Share the information related to the way Pega Call gets connected to the CTI environment with the telephony expert. The telephony expert is required to configure the CTI environment to allow Pega Call to connect and share the configuration details with the Pega Call implementation team.

Configuring computer telephony integration using Pega Call

To configure computer telephony integration (CTI) for Pega Sales Automation, you must enable the CTI functionality in the application and then add the `PegaCTI:CSR` access role to all the access groups for which you are providing CTI access.



Note: You must install and configure Pega Call to use CTI functionality in Pega Sales Automation. For more information, see the [Pega Call product page](#).

To configure your Pega Sales Automation implementation for CTI, complete the following steps:

1. In the header of Dev Studio, click **Configure > Org & Security > Groups & Roles > Access Roles**.
2. Select an access group.
3. On the **Definition** tab, in the **Available roles** section, click **Add role** and enter `PegaCTI:CSR`.
4. Click **Save**.
5. Repeat steps 2 through 4 for each access group for which you are providing CTI access.

Implementing Pega Knowledge

Pega Knowledge is a standalone application that is integrated with Pega Sales Automation. Pega Knowledge supports the creation and organization of rich content that can be suggested to sales representatives in the context of the selling activities that they are performing.

- [Pega Knowledge integration](#)
- [Adding Pega Knowledge records](#)

Pega Knowledge integration

Planning your application to support Pega Knowledge involves the following tasks.

Before you begin: Integration with Pega Knowledge is optional. If you plan to use this feature, you must activate it for your implementation. For more information, see [Activating application features](#).

1. Identifying the case types that need content. You can add knowledge content to the following case types:
 - Organization
 - Account
 - Contact
 - Opportunity
 - Lead
 - Household
2. Developing relevant content to support specific case types.

When planning, focus on the content that is necessary to support your sales processes. Write the content concisely and limit the number of Pega Knowledge articles so that your sales representatives do not have a large number of search results.

For additional information, see the *Pega Knowledge User Guide* on the [Pega Knowledge product page](#).

Adding Pega Knowledge records

You can add knowledge records to support the display of Pega Knowledge content to assist your sales representatives at any stage of the sales process. Pega Sales Automation includes a set of relevant knowledge records that the application automatically recognize when you associate knowledge articles with existing case types.

1. In the header of Dev Studio, click **Configure > Application > Inventory > Relevant Records**.
2. In the **Class name** field, select a case type to which you want to add a knowledge record.
3. Click **Add records**.
4. In the **Add KM Relevant Records** dialog box, enter the **Property name** and **Source class** for the knowledge record. The following property modes are supported:
 - STRING
 - PAGE
 - PAGE-LIST
 - PAGE-GROUP
5. Click **Submit**.

Implementing artificial intelligence

Pega Sales Automation artificial intelligence capabilities use predictive analytics, automated decisioning, and configurable business rules to provide insights and recommendations to sales representatives and sales managers throughout the sales cycle.

For more information, see [Artificial intelligence features in Pega Sales Automation](#).

- [Pega Workforce Intelligence integration](#)
- [Configuring Workforce Intelligence](#)

Pega Workforce Intelligence integration

Workforce Intelligence is a standalone application that is integrated with Pega Sales Automation. Workforce Intelligence delivers insight into your sales team's daily desktop usage patterns by collecting, aggregating, and synthesizing application utilization across the organization. You can use this data to analyze how work gets done across the enterprise—by humans and robots—and optimize application usage across all applications running on the desktop.

Before you begin: Integration with Workforce Intelligence is optional. If you plan to use this feature, you must activate it for your implementation. For more information, see [Activating application features](#).

To configure the Pega Sales Automation integration with Workforce Intelligence, you need the following information:

- The URL address, access token, and keystore file for the Workforce Intelligence server. This is used to establish the connection between Pega Sales Automation and Workforce Intelligence.
- The network ID for each Pega Sales Automation sales representative. Pega Sales Automation uses this ID to retrieve data from the Workforce Intelligence server.
- The email address for each Pega Sales Automation sales manager. Pega Sales Automation uses the sales manager's email address to determine which sales representatives to include in the data delivered to the sales manager.

For more information about Workforce Intelligence, see the [Workforce Intelligence product page](#).

Configuring Workforce Intelligence

Workforce Intelligence is a standalone application that is integrated with Pega Sales Automation.

To enable the Workforce Intelligence integration with Pega Sales Automation, complete the following steps:

- [Configuring settings for Workforce Intelligence](#)
- [Configuring authorization for Workforce Intelligence](#)
- [Updating the access group for the agent schedule](#)
- [Configuring Pega Sales Automation operators for Workforce Intelligence](#)

Configuring settings for Workforce Intelligence

To set up your Workforce Intelligence integration, you must enable the integration and then configure Pega Sales Automation to connect to your Workforce Intelligence URL.

1. In the Dev Studio header search text field, search for and select the `EnableWFI` dynamic system setting.
2. In the **Value** field, enter `true`.
3. Click **Save**.
4. In the Dev Studio header search text field, search for and select the `ConnectWFI URL` dynamic system setting.
5. In the **Value** field, enter your Workforce Intelligence system URL.
6. Click **Save**.
7. In the Dev Studio header search text field, search for and select the `D_WFIConfig` data page.
8. Click **Actions > Run**.

Configuring authorization for Workforce Intelligence

To set up your Workforce Intelligence integration, you must configure authorization protocols for Pega Sales Automation.

1. In the Dev Studio header search text field, search for and select the `SAWFIAccessToken` authentication profile.
2. In the **Client information** section, enter the **Identifier** and **Secret** received from the Workforce Intelligence server.
3. In the Dev Studio header search text field, search for and select the `SAWFI_OAuth OAuth 2.0` provider.
4. Enter the **Access token endpoint** URL for your Workforce Intelligence authentication profile.

5. In the Dev Studio header search text field, search for and select the `SAWFI_API` keystore.
6. Click **Upload File** > **Choose File** to browse for and select your Workforce Intelligence keystore file.
7. Click **Save**.

Updating the access group for the agent schedule

To set up your Workforce Intelligence integration, you must update the `PegaSA-WFI` agent schedule to use the access group of your implementation layer application administrator.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand **SysAdmin** and click **Agent Schedule**.
3. Search for and select the `PegaSA-WFI` agent schedule.
4. Click the **Security** tab.
5. In the **Access group** field, enter the access group for your implementation layer application administrator, for example, `SAPLus:SysAdmin`.
6. Click **Save**.

Configuring Pega Sales Automation operators for Workforce Intelligence

To set up your Workforce Intelligence integration, you must add Pega Sales Automation operators as users in Workforce Intelligence. You can then add Workforce Intelligence network IDs to your operator profiles in Pega Sales Automation. This allows the application to associate events and data with the correct operators.

For information about adding users in Workforce Intelligence, see the *Workforce Intelligence Administration Guide* on the [Workforce Intelligence product page](#).

After you add operators as users in Workforce Intelligence, complete the following steps:

1. In the header of Dev Studio, click **Launch web interface** > **SA for Sales Ops**.
2. In the navigation pane, click **Operators**.
3. Select the operator record that you want to update.
4. Click **Actions** > **Update Operator Profile**.
 - a. In the **WFI Network ID** field, enter the network ID from the user record in Workforce Intelligence.
 - b. In the **Email** field, verify that the email address matches the email address that you entered for the user in Workforce Intelligence.
5. Click **Submit**.

Implementing third-party integrations

You can enable several third-party application integrations if your business requires these tools.

- [Defining integration requirements](#)
- [Pega Sales Automation integration with Microsoft Exchange](#)
- [Configuring Pega for Outlook VSTO and Pega for Outlook Office](#)
- [Configuring listener accounts for emails and appointments](#)
- [Configuring Twitter integration](#)

- [Configuring Facebook integration](#)
- [Configuring LinkedIn integration](#)
- [Configuring Partner Relationship Management](#)

Defining integration requirements

You can integrate Pega Sales Automation with third party applications to extend the functionality of the application. Inventory the external applications that exist at your site and plan how to connect to them.

Some common integration points for Pega Sales Automation include interfaces to the following systems:

- Microsoft Exchange
- Microsoft Outlook (Pega for Outlook VSTO and Pega for Outlook Office)
- Social media
 - Twitter
 - Facebook
 - LinkedIn
- Customer applications

For information on integrating with customer applications, see [Integrating your application with external resources](#).

- Claims system
- Provider system
- Call center
- Product data
- Member data

1. Identify all integration points for your application.
2. Determine whether an interface already exists and how to connect to it.

Pega Sales Automation integration with Microsoft Exchange

Pega Sales Automation provides bi-directional calendar synchronization with Microsoft Exchange.

For more information, see [Email integration in Pega Sales Automation](#).

Configuring Pega for Outlook VSTO and Pega for Outlook Office

Pega Sales Automation provides two add-ins that you can use to integrate the application with Microsoft Outlook. The Pega for Outlook VSTO and Pega for Office Outlook add-ins support email integration between Microsoft Outlook and Pega Sales Automation and selective appointment integration between Microsoft Outlook and Pega Sales Automation.



Note: The Pega Sales Automation application installation includes the PegaFW-Outlook ruleset and SFAOutlookPortal.

Both add-ins provide real-time, context-specific insight into your Pega Sales Automation contacts, leads, opportunities, accounts, and organizations directly from Microsoft Outlook.

You can configure either Pega for Outlook VSTO or Pega for Outlook Office, but to leverage latest add-in technology, it is recommended to use Office add-in.

For more information, see [Email integration in Pega Sales Automation](#).

Configuring listener accounts for emails and appointments

With Pega Sales Automation, you can push emails and appointments from Microsoft Outlook (or another default mail client) to the Pega Platform. When an email or appointment is received, an email listener account identifies the recipients and maps the email or appointment to the appropriate operator.

You can associate incoming emails and appointments with contacts, organizations, and accounts in the application, and you can use Microsoft Outlook's vCard functionality to create a Pega Sales Automation contact from an Outlook contact.

Follow these steps to configure listener accounts for emails and appointments:

- [Creating the access group, operator, and email accounts](#)
- [Creating new email and appointment classes in the implementation layer](#)
- [Configuring associations with other application objects](#)
- [Updating the user interface](#)

Creating the access group, operator, and email accounts

To set up your listener account, you must create an access group and an operator ID, and configure email accounts to accept incoming emails and send bounce-back notifications.

1. In Dev Studio, create an access group. The default access group is `PegaCRM-SFA:Services` with the following attributes:

- `PegaRULES:User4` role
- Open, modify, search, and execute activity access to `PegaCRM-Document-Email`

If you do not use the default access group, you must change the **Service access group** on the `FWEmailCollector` service class. For more information, see [Access groups](#).

2. Create an operator role with the newly created access group. Enter this operator in the **Requestor Login** section for the email listener. For information, see [Operators](#).

3. Configure the `SyncEmailCalendar` email account.

This is the account that the email listener uses to import emails and appointments. You can link this account to any email address. For more information, see [About Email Account data instances](#).



Note: For Exchange integration, enter your Exchange server information when you configure the `SyncEmailCalendar` email account.

4. Configure the `SyncEmailCalendar` email listener.
5. Create or modify the default email account with the type **Notify**. Pega Sales Automation uses this account to send a bounce-back email if the incoming message cannot be associated with any object in the application. Configure the Notify email account to look like it is an official communication coming from the application.

Creating new email and appointment classes in the implementation layer

You must create new email and appointment classes in your implementation layer for the integration to work properly.

1. In Dev Studio, create new classes for emails and appointments:
 - a. Set the direct inheritance for the email class to `PegaCRM-Document-Email`.
 - b. Set the direct inheritance for the appointment class to `PegaCRM-Work-Appointment`.

For more information, see [Class layers](#).

2. Update the `PegaCRM-Document-Email.getAppointmentClass` decision table to default to your new appointment class.
3. Create a new service package using the `FWEmailCollector` service as a guide.
4. Update the service email rule to set the page class to your new email class.

Configuring associations with other application objects

You can associate emails and appointments with other objects in the application, such as organizations, accounts, and contacts. Because your implementation uses classes inheriting from Pega Sales Automation classes, you need to be sure that the email integration is associating emails and appointments with implementation layer objects.

For Emails, this association is configured through activities called from `PegaCRM-Embed-Recipient.lookupRecipient`. For appointments, this association is configured through activities called from `PegaCRM-Embed-Recipient.DoRecipientAssociations`.

1. To update contact associations, in Dev Studio, perform the following steps:
 - a. Override `@baseclass.crmFindContactByEmail`.
 - b. Replace references to the `PegaCRM-Entity-Contact` class in the `obj-browse` calls, pages, and classes to refer to your contact implementation class.
2. To update organization associations, perform the following steps:
 - a. Create an implementation class for `PegaCRM-Index-OrgDomains`.
 - b. Update the class references in `@baseclass.crmFindOrgByDomain`.

Updating the user interface

Emails and appointments appear in several places in the Pega Sales Automation user interface. You must update application rules to change class references to your implementation classes.

1. In Dev Studio, update the following rules for appointments:
 - a. `Data-Portal.SFAPortal_MyCalendar` - Modify the configuration of the `pxCalendar` control to update the class reference on the **Appointments** tab.
 - b. `@baseclass.crmPopulateAppointmentEvents` - Modify step 1 to update the report class.
 - c. `crmRelatedAppointments` - Override this rule in your implementation class to update the class reference to the report definition. Update both the repeating grid and the **Open Report** icon. There are versions in `PegaCRM-Entity-Contact` and `PegaCRM-Entity-Org`.
2. Copy the following report definitions from `PegaCRM-Work-Appointment` to your implementation class:
 - `crmAppointmentsByContact`
 - `crmAppointmentsByOperator`
 - `crmAppointmentsByOrg`
3. Update the following rules for emails:
 - `crmRelatedEmails` - Override this rule in your implementation classes to update the class reference to the report definition. Update both the repeating grid and the **Open Report** icon. There are versions in `PegaCRM-`, `PegaCRM-Entity-Contact`, and `PegaCRM-Entity-Org`.
4. Copy the following report definitions from `PegaCRM-Document-Email` to your implementation class:

- `crmEmailsByLinkedID`
- `crmEmailsByOperatorID`
- `crmEmailsByContact`

Configuring Twitter integration

Pega Sales Automation includes an integration with a Twitter API that retrieves information related to your organizations, leads, accounts, and contacts using a Twitter authorization profile.

Before you begin:

As part of this integration, you can call the REST APIs provided by Twitter using the XML response format. All the APIs used in the integration support Authentication, but they use only the Authenticated mode of requests. You must give users authorization to access Twitter from Pega Sales Automation. For Authentication, the application uses the Oath library.

Complete the following steps to configure your Pega Sales Automation integration with Twitter:

- [Setting up a Twitter account and application](#)
- [Configuring settings for Twitter integration](#)

Setting up a Twitter account and application

The Pega Sales Automation integration with Twitter requires an active Twitter account and application. For detailed information about how to complete each step, see the Twitter developer documentation.

1. Create a Twitter account.
2. Create a Twitter application to gather trend information.
3. Grant the Twitter application access to your Pega Platform application.

Configuring settings for Twitter integration

Before your sales representatives can use Twitter with Pega Sales Automation, you must configure your Twitter settings in App Studio. The settings required are provided when you set up your Twitter account and application.

1. In the navigation pane of App Studio, click **Settings**.
2. Click the **Social media** tab.
3. In the **Twitter integration** section, enter your Twitter settings as shown in the following table:

Setting name	Sample value
Access token URL	<code>https://api.twitter.com/oauth/access_token</code>
API URL	<code>https://api.twitter.com/1.1/</code>
Authorize user URL	<code>https://api.twitter.com/oauth/authorize</code>
Consumer key	<code>zLSxeSps7yT2InrSRdkEDw</code>

Setting name	Sample value
Consumer secret	P95m6E9wmiQJFVKDAg5nWlg5FsJxPAUXdOhAxRwNUZE
Request token URL	https://api.twitter.com/oauth/request_token

4. Click **Save**.
5. Refresh the `Declare_TwitterAppSettings` node level data page by completing the following steps:
 - a. In the Dev Studio header search text field, search for and select the `Declare_TwitterAppSettings` node level data page.
 - b. Click **Actions** > **Refresh**.

Configuring Facebook integration

You can integrate Pega Sales Automation with Facebook and then link Facebook profiles to leads, contacts, organizations, and accounts in the application.

To set up your Pega Sales Automation integration with Facebook, you must configure settings in the Pega Sales Automation application.

For additional information about how to integrate Pega Sales Automation with Facebook, see [How to post updates to Facebook from Pega Platform](#).

- [Setting up a Facebook account and application](#)
- [Configuring settings for Facebook integration](#)

Setting up a Facebook account and application

The Pega Sales Automation integration with Facebook requires an active Facebook account and application. For detailed information about how to complete each step, see the Facebook developer documentation.

1. Create a Facebook account.
2. Create a Facebook application to integrate with Pega Sales Automation.
3. Grant the Facebook application access to your Pega Platform application.

For additional information about how to integrate Pega Sales Automation with Facebook, see [How to post updates to Facebook from Pega Platform](#).

Configuring settings for Facebook integration

Before your sales representatives can use Facebook with Pega Sales Automation, you must configure your Facebook settings in App Studio. The settings required are provided when you set up your Facebook account and application.

1. In the navigation pane of App Studio, click **Settings**.
2. Click the **Social media** tab.
3. In the **Facebook integration** section, enter your Facebook settings as shown in the following table:

Setting name	Sample value
API URL	https://graph.facebook.com/v2.2/
Application ID	213682632091885
Application secret	4ed6d6cbd90c24df1c790c2f5b854de2

4. Click **Save**.
5. Refresh the `Declare_FacebookAppSettings` node level data page by completing the following steps:
 - a. In the Dev Studio header search text field, search for and select the `Declare_FacebookAppSettings` node level data page.
 - b. Click **Actions** > **Refresh**.

Configuring LinkedIn integration

To use the LinkedIn Sales Navigator directly in your Pega Sales Automation application, you must integrate the LinkedIn Sales Navigator with the Pega Sales Automation application. Integrating with LinkedIn Sales Navigator enables sales representatives to find new contacts with recommended leads and icebreakers, as well as to find commonality among sales representatives and their prospects. The LinkedIn Sales Navigator widget displays in contacts, opportunities, leads, and organization screens in Pega Sales Automation.

For more information, see [Configuring LinkedIn integration in Pega Sales Automation](#).

Configuring Partner Relationship Management

You can strengthen your channel relationships and optimize every step of the partner journey by using the Partner Relationship Management (PRM) functionality. For more information about PRM, see [Foster collaboration with Partner Relationship Management \(8.2\)](#).

To configure Partner Relationship Management, perform the following steps in your implementation layer.

 **Note:** Steps 8 - 14 of the following task are required only after upgrading from version 8.1.

1. Set the `UsePartnerAdministration` dynamic system setting to `true`.
2. Create a channel manager profile and configure it to have access to partners and partner territories.
3. Create partner organizations.
 - a. As a channel manager, log into Channel Manager portal and create partners by clicking **Create new partner** on the **Partner** tab.
 - b. Assign onboarding articles to your partner organization.
4. Create partner operators within the partner organization created above.
5. Set up collateral, by setting the `EnableKM` dynamic system setting to `true`.
6. Update the `OnboardingCategories` dynamic system setting with your onboarding categories to allow channel managers to add articles from these categories to the onboarding content.

 **Note:** For the steps below, *ImpAppName* is your application name.

7. In the Dev Studio header search field, search for and open the `crmAppExtModel` or `crmAppExtModelExtended` data transform. Perform the following steps:
 - a. Set the `.Prefix_PartnerAccessgroup` property to `ImpAppName`:PartnerAccessGroup.
 - b. Set the `.Prefix_PartnerAccessgroupWithSlash` property to `.Prefix_PartnerAccessgroup+"/"`.
 - c. Set the `.Role_PartnerSalesOps` property to `ImpAppName`:PartnerSalesOps.
 - d. Set the `.Role_PartnerSalesMgr` property to `ImpAppName`:PartnerSalesManager.
 - e. Set the `.Role_PartnerSalesRep` property to `ImpAppName`:PartnerSalesRep.
 - f. Set the `.Role_ChannelManager` property to `ImpAppName`:ChannelManager.



Note: `ImpAppName`:PartnerSalesOps, `ImpAppName`:PartnerSalesManager, `ImpAppName`:PartnerSalesRep, and `ImpAppName`:ChannelManager are the roles in the corresponding access groups of your application that are cloned from the product access groups.

8. **Optional:** Create dynamic system settings for the channel manager and partner access groups. For additional information, see [Creating a Dynamic System Setting](#).

Field	Channel manager	Partner sales rep
Owning Ruleset	PegaCRM-UserManagement	PegaCRM-UserManagement
Setting Purpose	<code>ImpAppName</code> AccessGroup/ ChannelManager	<code>ImpAppName</code> PartnerAccessGroup/ SalesRep
DSS Value	<code>ImpAppName</code> :ChannelManager	<code>ImpAppName</code> :PartnerSalesRep

Field	Partner sales manager	Partner sales ops
Owning Ruleset	PegaCRM-UserManagement	PegaCRM-UserManagement
Setting Purpose	<code>ImpAppName</code> PartnerAccessGroup, SalesManager	<code>ImpAppName</code> PartnerAccessGroup/ SalesOps
DSS Value	<code>ImpAppName</code> :PartnerSalesManager	<code>ImpAppName</code> :PartnerSalesOps

9. **Optional:** Create access groups for the channel manager and partner operators. For additional information, see [Creating an access group](#).

Field	Channel manager	Partner sales rep
Access group	<code>ImpAppName</code> :ChannelManager	<code>ImpAppName</code> :PartnerSalesRep
Clone it from OOTB access group	PegaCRM-SFA:ChannelManager	PegaCRM-SFA:PartnerSalesRep

Field	Partner sales manager	Partner sales ops
Access group	<i>ImpAppName</i> :PartnerSalesManager <i>ImpAppName</i> :PartnerSalesOps	
Clone it from OOTB access group	PegaCRM-SFA:PartnerSalesManager	PegaCRM-SFA:PartnerSalesOps



Note: If needed, include any roles, AROs, and privileges that are defined or introduced in the implementation layer.

10. **Optional:** Create dynamic system settings for the default channel manager and partner access groups. For additional information, see [Creating a Dynamic System Setting](#).

Field	Channel manager
Owning Ruleset	PegaCRM-UserManagement
Setting Purpose	<i>ImpAppName</i> DefaultUser/ SAPlus:ChannelManager
DSS Value	<i>ImpAppName</i> DefaultUserChannelManager

Field	Partner sales rep
Owning Ruleset	PegaCRM-UserManagement
Setting Purpose	<i>ImpAppName</i> DefaultUser/ SAPlus:PartnerSalesRep
DSS Value	<i>ImpAppName</i> DefaultUserPartnerSalesRep

Field	Partner sales manager
Owning Ruleset	PegaCRM-UserManagement
Setting Purpose	<i>ImpAppName</i> DefaultUser/ SAPlus:PartnerSalesManager
DSS Value	<i>ImpAppName</i> DefaultUserPartnerSalesManager

Field	Partner sales ops
Owning Ruleset	PegaCRM-UserManagement
Setting Purpose	<i>ImpAppName</i> DefaultUser/ SAPlus:PartnerSalesOps

Field	Partner sales ops
DSS Value	<i>ImpAppName</i> DefaultUserPartnerSalesOps

11. **Optional:** Log in as a Global Sales Ops and create access templates and templates for territories related to partners by performing the steps below:

- In the Tools gadget, click **Access templates** and use the *Access template naming convention* column of the table below as a base for creating access templates.
- Use the *Territory templates naming convention* column of the table below as a base for creating templates for all territories related to partners.

Access templates naming convention	Territory templates naming convention
<i>ImpAppName</i> DefaultUserChannelManager	<i>TerritoryID</i> + ~ + <i>ImpAppName</i> + DefaultUserChannelManager
<i>ImpAppName</i> DefaultUserPartnerSalesRep	<i>TerritoryID</i> + ~ + <i>ImpAppName</i> + DefaultUserPartnerSalesRep
<i>ImpAppName</i> DefaultUserPartnerSalesMgr	<i>TerritoryID</i> + ~ + <i>ImpAppName</i> + DefaultUserSalesMgr
<i>ImpAppName</i> DefaultUserPartnerSalesOps	<i>TerritoryID</i> + ~ + <i>ImpAppName</i> + DefaultUserSalesOps

12. **Optional:** Update the existing sales ops access group to include new roles and privileges. For additional information, see [Groups & Roles](#).
- Update the access group to *ImpAppName*:SalesOps.
 - Add roles from: PegaKM:TaxonomyManager, PegaKM:Approver, and PegaKM:Publisher.
13. **Optional:** If any ABAC policies are overridden in the implementation layer, update them with the corresponding classes. For additional information, see [Classes](#).
14. **Optional:** You can update the *OnboardingCategories* dynamic system settings to set onboarding categories as comma-separated values. Channel managers can share knowledge articles from these categories as onboarding documents for partners.
- [Configuring deal registration](#)
 - [Configuring Market Development Fund \(MDF\)](#)

Configuring deal registration

Deal registration is supported only for B2B deals.

To configure deal registration, perform the following steps:

1. **Optional:** To view and work on deals, add the *crmCanViewDeals* privilege to the channel manager and partner operator access roles in your implementation layer.
If your implementation role already has this product role configured as dependent, then this change is not required because it is covered through inheritance.

2. In the implementation lead class inheriting from the *PegaCRM-Work-SFA-Lead* class, expose *PartnerID* and *IsPartnerSourced* properties.
3. In the implementation opportunity class inheriting from the *PegaCRM-Work-SFA-Opportunity* class, expose *PartnerID* and *IsPartnerSourced* properties.
4. In your implementation layer, in the Lead case type, on the **Processes** tab, add the *DealSubmitForApproval* flow to the **Case wide supporting processes** list.

Configuring Market Development Fund (MDF)

As a channel manager, to approve and manage marketing activities initiated by partners you must configure Marketing Development Fund.

You must enable the *EnableMDF* and *MDFTimePeriodType* dynamic system settings.

1. In the header of Dev Studio, search for and open the *EnableMDF* dynamic system setting.
2. In the **Value** field, enter `true`.
3. Click **Save**.
4. In the header of Dev Studio, search for and open the *MDFTimePeriodType* dynamic system setting.
5. In the **Value** field, enter the time period for budgets, for example, `Quarter`.
6. Click **Save**.

Implementing the security model

Security planning involves defining authentication and authorization strategies for your application:

Authentication

Validates your identity.

Authorization

Determines the work objects you can access and the application functions you can perform.

For information about defining authentication and authorization strategies for your application, see the following topics:

- [Authentication schemes](#)
- [Authorization model](#)

Authentication schemes

The Pega Platform offers the following authentication types:

PRBasic

Based on passwords in the Operator ID data instances and the login form. This is defined by the HTML `@baseclass.Web-Login` rule, which your application can override.

PRSecuredBasic

Similar to PRBasic, but passes credentials by using Secure Sockets Layer (SSL) with Basic HTTP authentication. The login form is defined by the HTML `@baseclass.Web-Login-SecuredBasic` rule, which your application can override.

PRCustom

Supports access to an external LDAP directory or a custom authentication scheme.

PRExtAssign

Supports external assignments (Directed Web Access).

J2EEContext

Specifies that the application server in which the Pega Platform is deployed uses JAAS to authenticate users.

Implementing your authentication scheme

Your site can use a centralized, automated means of maintaining operator data instead of maintaining it manually in your application.

1. Discuss [Authentication schemes](#) with your site's security and application server teams.
2. Determine the appropriate authentication type.

For more information on authentication scheme planning, see [Authentication in Pega Platform](#).

Authorization model

The security authorization model determines user access privileges and work object permissions for the Pega Sales Automation application. Your security authorization model is based on the territory structure, operator IDs, and partners that you define for your sales team.

User authorization is based on both the operator ID privileges and Pega Sales Automation territory permissions.

Access to portals and work objects in the application is determined by operator ID privileges.

The ability to read, update, and create specific work objects is determined by the territory to which the work objects belong.

For more information about configuring territories and operators, see [Configuring the sales team structure](#).

- [Operator privileges by role](#)
- [Work object permissions](#)
- [Attribute Based Control \(ABAC\)](#)
- [Client-based access control \(CBAC\)](#)

Operator privileges by role

The operator role type determines which portals and work objects the operator can access.

Operator role	Description	Default portal
Developer	Manages the application. Modifies functionality and can launch any of the portals.	Dev Studio, SA for Sales Ops, SA Sales Manager, SA for Sales Rep
Sales Operations	Supports the sales team. Adds, modifies, and deletes	SA for Sales Ops

Operator role	Description	Default portal
	territories, operators, and partners, and accesses Sales Ops Tools to administer data.	
Sales Manager	Manages teams. Views management reports, dashboards, and forecasts, collaborates with team members, and accesses the team's opportunity pipeline and performance reports.	SA Sales Manager
Sales Representative	Manages sales. Communicates with prospects and customers, and works with the lead and opportunity pipelines.	SA for Sales Rep

Work object permissions

The application access privileges and territory permissions that you assign to operators in Pega Sales Automation determine how a user can interact with the work objects in the application.

- Operator privileges (role-based) give the user access to particular types of work objects in the application.
- Read, update, and create permissions for work objects are controlled by the territory that owns the work object.

For example, an operator with a Sales Representative role has access to opportunity work objects; however, to update an opportunity in the Northwest territory, you must grant the operator permission to update opportunity work objects in that territory.

- You can grant different levels of access to work objects within the same territory. For example, you can give a new operator read, update, and create access for lead and opportunity work objects in the Northwest territory, but only read access to organization objects in the same territory.
- A primary territory is defined and used as the default when new work objects are created. The owner of a work object has full access to the work object, regardless of territory access.

You can view the work object permissions assigned to your operators in the Sales Ops portal.

Attribute Based Control (ABAC)

Attribute Based Access Control (ABAC) controls row-level or column-level security through security policy rules available as part of the Pega Platform's ABAC feature.

For more information, see [Attribute-based access control](#) and [Sales Automation Attribute Access Control upgrade](#).

Client-based access control (CBAC)

Implementing client-based access control (CBAC) helps you satisfy the data privacy requirements of the European Union (EU) General Data Protection Regulation (GDPR) and similar regulations.

For more information about GDPR and CBAC, see [Supporting EU GDPR data privacy rights in Pega Infinity with client-based access control](#).

For more information about configuring CBAC in Pega Sales Automation, see CBAC section in the Pega Sales Automation Release Notes on the [Pega Sales Automation product page](#).

Modifying the user interface

You can change portions of the user interface to meet the needs of your business and users.

- [Designing for screen performance](#)
- [Implementing dashboards](#)
- [Reviewing the content of the portals](#)

Designing for screen performance

You want your application to respond immediately as your end users interact with customers. As you design your user interface, complete the following steps:

1. Review transactional service level agreements.
For example: The customer details must be available in one second or less while a representative is on the phone with the customer.
2. Review service performance, such as how long it takes back end systems to gather the data needed to display customer details.
3. Review data requirements to determine which fields contain the information that the representatives absolutely need for the task that they need to perform.
4. Determine if any network configuration could cause latency.
For example: Representatives who are logged in through a corporate VPN or in a remote location might cause an issue.

Implementing dashboards

When you first access your application portal, you see the dashboard, which typically contains a navigation panel on the left.

- [Determining which part of the user interface to modify](#)
- [Portal dashboards](#)
- [Customizing dashboards](#)

Determining which part of the user interface to modify

You can customize the content of the dashboard so that it displays any data that you integrate into your application.

Portal dashboards

Pega Sales Automation provides custom portals and configurable dashboards for supported operators.

Portals provide a comprehensive workspace while dashboards centralize key information managed in the portal. Access to portals and their corresponding dashboards is determined by the user's access group and operator ID.

Determine the appropriate dashboard information to display for each of the following Pega Sales Automation portals:

- SA for Sales Ops
- SA for Sales Rep
- SA Sales Manager

Defining the dashboard layouts for portals

The application includes configurable, role-based dashboards for each portal. A Sales Ops operator can publish default dashboard layouts for Sales Operations, Sales Manager, and Sales Representative operators. Individual operators can modify the default dashboard layouts, as needed.

To identify the default dashboard components that meet your business needs, perform the following steps:

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
2. In the navigation pane, click **Tools** and select a dashboard that you want to modify.
3. In the upper right corner of the dashboard, click the **Personalize the dashboard** icon.
 - a. Review the available templates for the dashboard.
 - b. In any slot, click **Add widget(s)** and review the available widgets for the dashboard.
4. Determine which template and widgets to use for the dashboard display.

Customizing dashboards

A Sales Ops operator can publish default dashboard layouts for Sales Operations, Sales Manager, and Sales Representative operators. Individual operators can modify the default dashboard layouts, as needed.

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
2. In the navigation pane, click **Tools** and select the dashboard that you want to modify.
3. In the upper right corner of the dashboard, click the **Gear** icon.
4. Click **Switch template** and select a template for the dashboard.
5. Click **Add widget(s)** in any slot in the template.
 - a. Select the widgets that you want to display on the dashboard.
 - b. Click **Add selected**.
6. Drag and drop widgets from one slot to another to change their position on the dashboard.
7. Click **Publish**.
8. Repeat steps 3 through 7 for other dashboards, if necessary.

For more information about customizing dashboards, see [Pega Sales Automation dashboard customization](#).

Reviewing the content of the portals

Review the content of your application's portals to ensure that only relevant data is presented.

1. In the header of Dev Studio, click **Launch web interface > SA for Sales Ops**.
 2. Review each of the portal sections.
 3. Repeat this process for the **SA for Sales Rep** and **SA Sales Manager** portals.
- [Locating the user interface element to change](#)
 - [Adding properties to sections](#)
 - [Adding sections](#)
 - [Changing section visibility](#)
 - [Designing an application skin and styles](#)
 - [Updating the skin](#)

Locating the user interface element to change

You can customize the content of the dashboard so that it displays any data that you integrate into your application. To determine which user interface element to modify to implement your change, use the Live UI feature of the Pega Platform.

1. In Dev Studio, launch the portal that you want to modify.
2. Click the **Open runtime toolbar** in the bottom right corner.
3. Click the **Live UI** icon.
4. Select an element to observe and click the **Display info panel for this item** icon.
5. To update the rule, click the **Open rule in Dev Studio** icon and make your changes.

Adding properties to sections

Pega Sales Automation uses standard Pega Platform user interface components, so the update process is the same as any Pega Platform application.

1. In Dev Studio, launch the portal that you want to modify.
2. Click the **Open runtime toolbar** in the bottom right corner.
3. Click the **Live UI** icon.
4. Select the section that you want to modify.
5. To open the rule in Dev Studio, click the **Open rule in Dev Studio** icon.
6. In the navigation panel of Dev Studio, click **App**.
7. Locate the property that you want to add to the section and drag it into that section.
8. Click **Save as** and then save the ruleset to your application.

Adding sections

Pega Sales Automation uses standard Pega Platform user interface components, so whether you are updating the dashboard, portals, or the composite, you do so in the same way as any Pega Platform application.

Pega Sales Automation uses standard Pega Platform user interface components. Whether you are updating the dashboard, Patient 360 view, or the portals, you follow the same procedure as you do for any Pega Platform application

1. In Dev Studio, launch the portal that you want to modify.
2. Click the **Open runtime toolbar** in the bottom right corner.
3. Click the **Live UI** icon.
4. Locate the area where you want to add the section.
5. In the navigation panel of Dev Studio, click **App**.
6. Locate the section that you want to add and drag it into your specified section.
7. Click **Save as** to save the rule to the ruleset for your application.

Changing section visibility

Use When rules to determine which layouts and sections are visible. For example, if you are a manager, modify the visibility of a section to ensure that only a manager can see the content of the section.

1. In Dev Studio, access the section for which you want to change visibility.
2. Click anywhere inside the section and then click the **View properties** icon.
3. From the **Visibility** drop-down list, select **Condition**.
4. Enter a When rule or a conditional expression.
5. Click **OK** and then click **Save**.

Designing an application skin and styles

Your site very likely has user experience (UX) standards for the appearance of your applications. Your task is to determine which styles have to change to adhere to those standards.

1. In the header of Dev Studio, click **Configure > User Interface > Skins, interfaces, & templates**.
2. Click the **Component styles** tab.
3. Review each component.



Note: As you plan your application styles, consider using mixins to provide incremental style changes. For more information, see [Using mixins in the skin to drive consistency and efficiency](#).

Updating the skin

You can update the look and feel of Pega Sales Automation to reflect the color scheme of your organization. The application uses the standard Pega Platform skinning features in your application.

To access Pega-provided formats from Pega Platform releases, the `SAEndUser` skin rule for Pega Sales Automation inherits from the `pyEndUser` skin rule from the Pega UI Kit.

For more information about skins, see [Creating a skin](#) and [Skin inheritance](#).

1. In Dev Studio, click *Application Name* > **Skin**.
2. Update the options as needed, and then click **Save as**.
3. If you give your skin a different name, update the application to reflect it by performing the following steps:
 - a. In Dev Studio, click *Application Name* > **Definition**.
 - b. Change the value in the **Skin** field to the name of the skin that you created.
4. Click **Save**.

Testing a new application

Testing a new application involves testing in different environments.

- [Testing your application in the Build environment](#)
- [Testing in the test or production environments](#)
- [Testing in the UAT environment](#)

Testing your application in the Build environment

Test a new application in the Build environment before migrating the application to a test or production environment. Testing in the Build environment enables you to verify that basic functionality and interfaces work correctly and that performance is acceptable.

1. Run functional tests to test specific features from the end-user perspective.
2. Verify that the Pega-provided reports and your custom reports run successfully, and that they show your implementation layer data, rather than the default demonstration data. This can be an automated test.
3. Test all integrations, both independently and with associated integrations.
Test integrations for any optional Pega Sales Automation components and other applications that you plan to use. See the product documentation for the component or application to determine which product components to test.
4. Test security. Test the most common roles to ensure that the required access groups are configured and point to the correct software version.

Testing in the test or production environments

After you import the application to a test or production environment, test the application in the new environment to verify that it works correctly in that environment.

1. Verify that the source and the destination files are the same.
2. Run functional tests to test specific features from the user perspective.
3. In the test or production environment, run the Application Guardrails Compliance Score to ensure that the application meets guardrails.
4. Verify that the Pega-provided reports and your custom reports run successfully, and that they show your implementation layer data, rather than the default demonstration data. This can be an automated test.
5. Test all integrations, both independently and with associated integrations.
Test integrations for any optional Pega Sales Automation components and other applications that you plan to use. See the product documentation for the component or application to determine which product components to test.
6. Verify that the integrations point to the correct system of record, and not to the system of record for the Build environment.

7. Test security. Test the most common roles to ensure that the required access groups are configured and point to the correct software version. Use these common roles in your smoke tests (see next step).
8. Run a smoke test to compare the source and destination environments. Verify that all tests that pass in the Build environment also pass in the test or production environment. If anything fails, compare the environments to determine whether a difference in environment could cause the test to fail. If the environment caused the failure, either fix the issue that caused the failure or adjust the test as appropriate for the new environment.
9. Run performance tests to verify that performance meets expectations. Pega recommends automated performance testing. Save the results so that you can compare them to future performance test results to determine whether an application update has a performance impact.

Testing in the UAT environment

After you complete testing in a Test environment, it is common to perform User Acceptance Testing (UAT) in a designated UAT environment, which could be a pre-production environment. UAT ensures that users will be able to successfully complete work and meet business objectives.



Note: Organizations that use Scrum for application development will complete less formal UAT as part of each sprint cycle.

1. Verify the integrity of the UAT environment.
2. Have the end-users (or business analysts acting the role of end-users) run scripts to test all scenarios including boundary and exception testing. The end-users (that is, the trainers, managers, and directors), perform the following steps during UAT:
 - a. Verify that there are no major issues.
 - b. Review changes to understand the features.

Packaging a new application

To migrate a new application to a different environment, you must first package the application so that it can be imported to the new environment.

- [Merging application changes](#)
- [Packaging an application for migration](#)
- [Importing the packaged application](#)

Merging application changes

If you developed your application features in separate branches, use the **Merge Branches** wizard to merge the branches before you package the application. The wizard shows any merge conflicts so that you can correct them before you merge the branches.

Packaging an application for migration

Before you can migrate a new application to a different environment, you must package the relevant data instances and rulesets into a product rule. The product rule is an instance of `Rule-Admin-Product`, and it is referred to as the RAP file.

1. In the header of Dev Studio, click **Configure > Application > Distribution > Package** to start the Application Packaging wizard. For information about using the wizard, see [Product rules](#).
2. Complete each page of the **Application Packaging** wizard.
3. On the last page of the wizard, click **Preview**.
4. Review the contents of the generated RAP file.
5. On the last page of the wizard, click **Modify** to make any changes.
6. When the RAP file is complete, click **Export**.
The wizard creates a .ZIP file in the **ServiceExport** directory on the current application server node.

Importing the packaged application

To deploy a new application to a different environment, import the .ZIP file that contains the packaged application to the new environment.

1. In the header of Dev Studio, click **Configure > Application > Distribution > Import**.
2. Use the Import wizard to import the target .ZIP file. For more information, see [Importing rules and data from an archive by using a wizard](#).

For information about how to swap the database connection pointers to your production database after an import to a production environment, see the Pega Platform Upgrade Guide at [Deploy Pega Platform](#).

Production maintenance and monitoring

Production maintenance and monitoring include the following procedures:

- [Business rule maintenance in the Production environment](#)
- [Application health monitoring](#)
- [Identifying and reporting issues](#)

Business rule maintenance in the Production environment

As a manager, you can update **Dialogs**, and **Coaching Tips** in the production environment by using the **Tools > Configuration** option available in each process action of the case type. This feature allows managers to change these options without having to wait for a rule deployment. Managers must have the same privileges associated with the PegaCA:Manager role in order to maintain these rules.

You can give managers the ability to update other rule types in the production environment. For example, managers can update the Goals and Deadline for a certain case type. These rules must be delegated in Dev Studio first. For more information on rule delegation, see [Delegating a rule or data type](#).

Application health monitoring

[Pega Autonomic Event Services](#) is an application that automatically monitors, retrieves, and organizes the alert data from one or more clustered systems throughout the enterprise. Pega also provides the [Pega Predictive Diagnostic Cloud](#), which allows you to benefit from Pega Autonomic Event Services without installing it locally. Pega Predictive Diagnostic Cloud is a Software as a Service offering of Pega Autonomic Event Services.

Implementing the following best practices in your application can help to ensure optimal response times and overall application health:

- Segment application agent processing to a dedicated JVM (Java Virtual Machine). This configuration ensures that end users do not have to share resources with background processes.
- Monitor integration response time. Over time, slow integration points can cause average handle times to increase. When queues start to grow, it becomes very difficult to recover, which might require usage of offline services or a backup application.

Identifying and reporting issues

As with any application, your users will encounter issues that they need to report as they begin to use the application. When deploying your application to the production environment, complete the following steps:

1. Identify operational staff who will be responsible for responding to issues reported in the production environment.
2. Establish procedures with those resources to triage, respond to, and escalate issues.
3. Determine procedures for delivery of changes to the production environment.

