



# Pega Government Platform

## 8.8

4 March 2024

# CONTENTS

<b>Get started with Pega Government Platform</b>	<b>8</b>
<b>Release notes and patches</b>	<b>9</b>
Release notes.....	9
Pega Government Platform 8.8 New features.....	10
Time Management.....	10
Item Entity.....	10
Pega Government Platform 8.8 enhancements.....	11
Other enhancements.....	13
Upgrading from prior versions.....	14
Issues addressed in this version.....	14
Known issues in this version.....	15
Rule changes in Pega Government Platform 8.8.....	17
Upgrading from earlier versions of Pega Government Platform.....	18
Required hotfixes.....	18
Deprecated and withdrawn rules.....	18
Deprecated rules.....	19
Withdrawn rules.....	28
Patch.....	53
8.8.1 patch release.....	54
<b>Install</b>	<b>55</b>
Completing the prerequisite tasks.....	55
Backing up your system.....	55
Installing the application.....	56
<b>Update and hotfixes</b>	<b>63</b>
Update.....	63
Completing the prerequisite tasks.....	63
Backing up your system.....	63
Updating the application.....	65
Hotfixes.....	84

Hotfixes for Pega Government Platform 8.x.....	85
Pega Government Platform 8.8.....	85
Pega Government Platform 8.7.....	86
Pega Government Platform 8.6.....	86
Pega Government Platform 8.5.....	88
Pega Government Platform 8.4.....	89
Pega Government Platform 8.3.....	91
Pega Government Platform 8.2.....	92
Pega Government Platform 8.1.....	92
Hotfixes for Pega Government Platform 7.x.....	94
Pega Government Platform 7.4.....	94
Pega Government Platform 7.31.....	96
Pega Government Platform 7.21.....	97
Pega Government Platform 7.16.....	98
<b>Implement.....</b>	<b>101</b>
Preparing for the implementation.....	102
Testing a new application.....	111
Packaging an application.....	114
Production maintenance and monitoring.....	116
Application data model.....	117
Pega Government Platform entities.....	118
Person entity.....	120
Business entity.....	129
Vehicle entity.....	140
Household entity.....	150
Facility entity.....	159
Item entity.....	172
Entity attachments.....	184
Entity History and Field Audit.....	186
Extend History and Field audit to a new entity.....	188
Extend field audit to a new field of an existing entity.....	189
Entity merge.....	189

Creating New Entities.....	193
Implementing Pega Government Platform features.....	208
Review Checklist.....	209
Consent Agreement.....	213
Document intake.....	217
Verify document.....	221
Nearby locations.....	223
Evaluate.....	228
Activity plan.....	232
Assessments.....	247
Multiple entity intake.....	251
Case report.....	258
Programs.....	264
Operator Case Involvement (Touch) reporting.....	274
Pega Government Platform case types.....	279
Initiate event.....	280
Investigative Case Management.....	285
Interview case.....	291
Evidence case.....	292
Application request.....	295
Dashboard.....	297
Reports on Investigation.....	298
Reports on Event.....	298
Reports on Application request.....	300
Nearby cases chart.....	301
Add a new report/chart to dashboard.....	302
Search program Landing Page.....	303
Adding a search field.....	303
Adding field in search results.....	303
Adding field to Search criteria.....	304
Manage associations.....	304
Manage associations case architecture.....	304

Association validations.....	306
Adding a data element.....	307
Adding an entity type.....	307
Auditing an Association.....	308
Extend Audit History for an Association.....	308
Time Management.....	309
Adding new fields to project tables.....	311
Adding, deleting, or updating records in a project table.....	313
Adding new fields to Time code data objects.....	315
Adding, deleting, or updating records in Time code data objects.....	316
Adding new fields to the time entry view.....	317
Adding new fields to time entry log data objects.....	318
Changing the due time of a time sheet.....	320
Time entry flow.....	320
Appendix A: Pega Government Platform Application overview.....	322
Appendix B: Creating new entities.....	331
Entities.....	336
New entity overview.....	337
Development steps.....	338
Rule inventory.....	348
Entity data type rules.....	351
Appendix C: Mobile Strategy.....	353
Pega Mobile overview.....	353
Pega Government Platform Mobile.....	355
Mobile Offline.....	356
Changes for Mobile Offline.....	358
Changes to Pega Government Platform features.....	362
Pega JavaScript API.....	363
Limit the Visibility of sensitive data items through attribute-based access control (ABAC).....	363
Leverage Pega Government Platform from Pega Customer Service.....	369
Pega Government Platform and Pega Customer Service Implementation..	370

Business Objective.....	370
Purpose.....	370
Solution.....	370
Implementation steps.....	371
Integrating Industry Applications with Pega Customer Service.....	372
Case Study: Extending Pega Government Platform with Pega Customer Service	
.....	374
Overview.....	374
Driving UI with Local Data.....	375
Option Map.....	375
Architecture.....	376
Using OptionMap Data in a Picker control.....	378
<b>Product Overview</b>	<b>379</b>
Application intake features.....	379
Manage information features.....	382
Case Reviewer features.....	385
Association Management Framework.....	390
Entity Profile views.....	391
Investigative case management.....	393
Pega Government Platform case types and workflows.....	399
Example case types.....	399
Standard case types.....	402
Roles, portals, and dashboards in Pega Government Platform.....	407
Primary data entities.....	411
Time Management.....	415
Time entry.....	415
Manage Time.....	416
Roles in Time Management.....	417
Further reading.....	417
<b>Additional</b>	<b>418</b>
Pega Certification and Licensing for Government hotfixes.....	418
Hotfixes for Pega Certification and Licensing for Government 8.x.....	418

Hotfixes for Pega Certification and Licensing for Government 7.x. .... 420

# Get started with Pega Government Platform

Explore the links below to access product documentation and other useful information for Pega Government Platform

## Learn about

- Release notes
- Patch

## Setup

- Install
- Upgrade
- Implement

## Use

- Hotfixes
- Product overview
- Deprecated and withdrawn rules

# Release notes and patches

- [Release notes](#)
- [Patch](#)

## Release notes

Pega Government Platform (PGP) 8.8 provides an array of new capabilities and enhancements that caters to accelerate the development of Government use cases in DOD defense based processes, program management including grants, licensing, certificates, benefit programs and investigative case management.

Features and efforts comprise of time management components, new item entity, enhanced associations, activity plan, program configuration and application request management.

Pega Government Platform 8.8 includes an updated user experience based on Pega Platform 8.8 functionality.

For more information and a list of additional documents available for this release, see the [Pega Government Platform product page](#).

For a complete list of new features, see the [Release Notes](#).

- [Pega Government Platform 8.8 New features](#)
- [Pega Government Platform 8.8 enhancements](#)
- [Other enhancements](#)
- [Upgrading from prior versions](#)
- [Issues addressed in this version](#)

- Known issues in this version
- Rule changes in Pega Government Platform 8.8
- Upgrading from earlier versions of Pega Government Platform
- Deprecated and withdrawn rules

## Pega Government Platform 8.8 New features

The Pega Government Platform 8.8 release includes a number of new features.

- Time Management
- Item Entity

### Time Management

The Time Management application provides a generalized time management system for tracking and reporting time for case workers that is consistent with applicable standards and guidelines.

Users can use this case to capture their hours worked, submit time entries for approval, and re-submit rejected time entries on a daily basis. A user has multiple opportunities to record their time throughout the workday, or can enter their time for the entire day and submit it for approval as part of their end-of-day activities.

Time Management will also feature a **Manage time** landing page, where users can view their time entry logs for a given time period and managers can view all their team's time entry logs.

### Item Entity

Pega Government Platform provides data support to collect data about items in the form of the Item Entity. The Item Entity feature contains a persistent datastore and three supporting case types that manage the Item Entity datastore. Item Entity also

includes information such as manufacturer, owners, subitems, and so on. Case workers can also use the Copy item tool to create similar items.

## Pega Government Platform 8.8 enhancements

The Pega Government Platform 8.8 release includes a number of product enhancements.

### Enhanced ICM Report of Investigation

Using the case report component, you can now generate a report of a case which contains the data that is captured using various other components along with external organization as a part of case summary. One can also include digital signature in the report, both in pdf and word format, as per requirement.

You can now generate a Report of Investigation in both PDF and Microsoft Word formats. You can display data, such as subject, evidence, and assessments from various sections in a tabular format. The ROI component is flexible, allowing you to choose to include data, such as evidence, subjects, interviews, and assessments from various sections to be a part of the end report.

### Application request

Application request case type now also supports applying for a program or any of its descendants, for example a license, for an item entity.

### Activity Plan

Activity Plan Usage is the new feature added to the activity plan template. In Activity plan usage you can see how many times an existing plan template or a custom template is used. It also provides a feature to convert custom plan template to plan templates with a click of a button.

## Associations

Associations now supports date range feature (association duration) for each entity or profile which is added as an association. You can now add the same entity or profile as an association with different date ranges. The preview and visualization of associations also shows the duration of association.

## Program configurator

Through Program configurator you can create programs with an image. This image is shown in license case, application and even in report of investigation (ROI).

## Evidence

As a User, you can now capture an item entity, can create a new item or use existing item entity, as a part of evidence.

## Dashboards

Application request reports are added to the dashboard in addition to existing reports. Application request by status report provides summarized information of the application request cases with their case status. Recent programs issue report provides a high-level summary of how many different programs has been issued for each day in the past 7 days.

## Entity Profiles

All entity profiles now support the feature of auditing history, this includes the case details where the entity has been used or edited and auditing changes made to the primary fields like name and address.

In ICM User can now initiate subject from entities like person, business, vehicle, and facility. Both ICM and PGP users can initiate events from person profile.

## Enhance Entity Components Performance

Entity flow components are persisted to the system only if changes are made to the entity. This increases performance by reducing the number of persist operations

## Other enhancements

The Pega Government Platform 8.8 release includes minor enhancements that improve accessibility and ease-of-use.

### Support for security policies in User Registration

Business users can now enforce security policies for user registration. They can choose wide range of settings like length of password and alphanumeric validations.

### Confirmed Localization readiness

All the applications shipped are verified for localization readiness.

### Elastic search SRS adoption

Search and reporting service (SRS) is an independent micro-service that provides full-text search capabilities and is a multi-tenant and cloud-based service. The SRS is independent of platform version, and it is convenient for indexing class specific records. The current developed works in both Embedded Elastic search and in SRS environment for entities like person, vehicle, business, facility and ICM cases like Investigation and subject. .

### Mashup ready harness

Pega Government Platform provides mashup ready harnesses like PersonalDetails and BusinessDetails to display person and business details in the mashup when referred.

## Optimized ID generations for data object

The process of generating unique ids for data objects has been optimized, with which performance has improved.

## Internationalization of Date placeholders

Pega Government Platform contains many screens where users can enter details according to their need. For the Date inputs now, we are providing a place holder in such a way that it changes based on the operator's time zone.

## Offline Mobile feature

Users can use some of icm application features in mobile even though device does not have internet connection. User can create cases like Interview and Evidence both as standalone and from investigation.

## Upgrading from prior versions

Pega recommends that you avoid using some use cases during an upgrade. For a list of common use cases to avoid for any Pega application, see [Common use cases to consider during an upgrade](#).

## Issues addressed in this version

This section describes issues resolved in this release that are of the most interest to and are likely to have the most impact on the Pega user and developer community.

For each issue, a reference number is provided, and the prefix of the reference number indicates the issue type. You can use the reference number of an issue in your related conversations with [Pega Support](#).

The release notes include the following issue types:

### INCs

Customer-reported incidents. For example, INC-183895.

**SRs**

Support requests, which were used instead of incidents in older releases. For example, SR-D79601.

**ISSUEs**

Pega-identified issues. They might or might not be related to customer-reported incidents. For example, ISSUE 654263 (which might also be written as just 654263).

**SEs**

Sustenance engineering activities. For example, SE-60265.

Starting Q2 2021, all customer-reported issues are logged as INCs. You can view INCs that you logged in the [My Support Portal](#). INCs logged by other Pega customers, and all other issue types (SR, ISSUE, and SE), are available in Pega internal tracking systems, in addition to these release notes.

ID	Title and Description
ISSUE 698104	No access to report error when work indexing is enabled



**Tip:** You can look up Pega Platform resolved issues in the [Pega Platform Resolved Issues](#).

## Known issues in this version

This table describes the known issues that might occur in this release.



**Note:** To submit new issues or find out more about known issues, or to request a hotfix, go to the [Pega Product Support Community](#). Look up or subscribe to your Support Requests (SRs) in [My Support Portal](#). Ensure that you refer to the issue ID (SR, BUG, or FDBK) in all communications.

<b>ID</b>	<b>Title and Description</b>	<b>Suggested Resolution</b>
ISSUE 625115	Empty assignment key error on click of edit in investigation case, for user with no edit access rights.	Override corresponding section and have a when condition.
ISSUE 676905	Unable to open the tab review checklist tab in application request unless case is refreshed.	Refresh the case to view the details.
ISSUE 715417	Unable to create Event case from Dashboard page.	
ISSUE 730107	"Invalid report content page is specified" error using datapage in ABAC.	
ISSUE 716163	Localization issue in Tracksecuritychanges.	
ISSUE 737955	Home page worklist page is not showing all assigned cases	
ISSUE 89663	Case insensitive not working for embedded elastic search	
ISSUE 741577	On assignment save, document attachments are not being displayed : In Application request case, in Document intake screen, if attachments are added to any document type (except to first row), on clicking save button, added attachments become invisible. Same is the case for the document section present in optional process- Request more information.	

ID	Title and Description	Suggested Resolution
	<p> <b>Note:</b></p> <p>We can still see the attached documents under Documents tab.</p>	
ISSUE-7644 97		

## Rule changes in Pega Government Platform 8.8

Pega Government Platform has withdrawn a few rules during this release to better maintain the product and make it easier for you to extend the rules when needed.

### Withdrawn rules

Rule type	Rule name	Ruleset	Comments
TextFile	Bootstrap.css	PegaPS	Dead code as its not being used in End user portal.
TextFile	Custom.css	PegaPS	Dead code as its not being used in End user portal.
TextFile	ProgressIndicator.js	PegaPS	Dead code as its not being used in the ProgressIndicator control.

## Other rule changes

Rule type	Class	Rule name	Ruleset	Comments
Activity	Code-Security	IACAuthentication	PGPOVERRIDE	Mandated the verification of password at step 6.
Text file		Vis JS	PegaPS	Updated js with the latest js from Vis-Network and this is deprecated. Use Js by referring to the URL. Refer EntityAssociationControl.

## Upgrading from earlier versions of Pega Government Platform

This section provides information for customers upgrading to Pega Government Platform 8.8 from an earlier release.

- **Required hotfixes**

## Required hotfixes

For a list of required hotfixes to apply when updating from previous versions, see [Pega Government Platform hotfixes](#).

## Deprecated and withdrawn rules

Pega Government Platform has deprecated some rules during the releases to better maintain the product and make it easier for you to extend rules when needed. After two releases these rules will be withdrawn as a practice.

The following table lists the count of deprecated and withdrawn rules for every release in Pega Government Platform.

Release	Deprecated Rules	Withdrawn Rules
8.8	2	4
8.7	7	67
8.6	4	48
8.5	35	165
8.4	0	31
8.3	13	37
8.2	5	34
8.1	3	2

- [Deprecated rules](#)
- [Withdrawn rules](#)

## Deprecated rules

This page lists the list of deprecated rules of Pega Government Platform for 8.x releases.

### 8.8 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.8 release.

Name	Ruleset	Ruleset version	Applies to
webwb Vis JS	PegaPS	08-08-01	
pyStartCase	PegaPS	08-08-01	PegaPS-Work-Merge

## 8.7 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.7 release.

Name	Ruleset	Ruleset version	Applies to
CaseActionArea	PegaPS	08-07-01	PegaPS-Work-Question
CaseActionAreaButtons	PegaPS	08-07-01	PegaPS-Work-Question
CancelFlow	PegaPS	08-07-01	PegaPS
LogOffPS	PegaPS	08-07-01	PegaPS
PostConfirmCancel	PegaPS	08-07-01	PegaPS
PostEvidence	PegaPS	08-07-01	PegaPS-Work
pyStartCase	PGPCosmos	08-07-01	PegaPS-Work-Merge

## 8.6 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.6 release.

Name	Ruleset	Ruleset version	Applies to
ListCount	PegaPS	08-06-01	
CancelFlow	PegaPS	08-06-01	PegaPS

Name	Ruleset	Ruleset version	Applies to
LoadClassContext	PegaPS	08-06-01	PegaPS-Data-Context-Class
pyStartCase	ICMCosmos	08-06-01	PegaPS-Work-RequestAccess

## 8.5 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.5 release.

Name	Ruleset	Ruleset version	Applies to
QuestionPageLabel	PegaPS	08-05-01	Embed-PegaQ-Results-QuestionGroup
.QuestionPageLabel	PegaPS	08-05-01	Embed-PegaQ-Results-QuestionGroup
PGPSampleData 8.3	PegaPS	08-05-01	
PGPSampleData 8.4	PegaPS	08-05-01	
PGPSampleDataSCs 08.10	PegaPS	08-05-01	
D_GetQuestionGrou pLabel	PegaPS	08-05-01	Rule-PegaQ-QuestionGroup
ListCount	PegaPS	08-05-01	
RelatedEntities	PegaPS	08-05-01	PegaPS-Data-Entity
ManageTeamMobil e	PegaPS	08-05-01	PegaPS-Work
HeaderInformation	PegaPS	08-05-01	PegaPS-Data-Entity-Business

Name	Ruleset	Ruleset version	Applies to
ViewDetails	PegaPS	08-05-01	PegaPS-Data-Entity-Business
FacilityLocation	PegaPS	08-05-01	PegaPS-Data-Entity-Facility
AddAttachments	PegaPS	08-05-01	PegaPS-Data-Entity-Vehicle
AddRelations	PegaPS	08-05-01	PegaPS-Work
RelatedPersonnel	PegaPS	08-05-01	PegaPS-Work
AddAssociates	PegaPS	08-05-01	PegaPS-Work-Manage-Associate
pyCaseContent	PegaPS	08-05-01	PegaPS-Work-Manage-Education
pyCaseContent	PegaPS	08-05-01	PegaPS-Work-Manage-Experience
CaseActionArea	PegaPS	08-05-01	PegaPS-Work-Question
CaseActionAreaButtons	PegaPS	08-05-01	PegaPS-Work-Question
CancelFlow	PegaPS	08-05-01	PegaPS
LogOffPS	PegaPS	08-05-01	PegaPS
PostConfirmCancel	PegaPS	08-05-01	PegaPS
pyStartCase	PegaPS	08-05-01	PegaPS-Work-Merge
ViewDetails	PegaPS	08-05-01	PegaPS-Data-Entity-Business
InitiateEvent	PegaPS	08-05-01	PegaPS-Work-Event-Initiate

Name	Ruleset	Ruleset version	Applies to
InitiateAskQuestion	PegaPS	08-05-01	PegaPS-Work-Question
GovtServicesDetails	PegaPSPortals	08-05-01	PegaPS-Work
pyCaseContent	PegaPSUserServices	08-05-01	PegaPS-Work-User-ChangePassword
ForgotPassword	PegaPSUserServices	08-05-01	PegaPS-Work-User-ForgotPassword
Registration	PegaPSUserServices	08-05-01	PegaPS-Work-User-Registration
pyStartCase	PGPCosmos	08-05-01	PegaPS-Work-Entity-Facility-Add
pyStartCase	PGPCosmos	08-05-01	PegaPS-Work-Manage-Branch
pyStartCase	PGPCosmos	08-05-01	PegaPS-Work-Manage-Member
pyStartCase	PGPCosmos	08-05-01	PegaPS-Work-Merge

## 8.4 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.4 release.

Name	Rule type	Ruleset	Ruleset version	Applies to
SaveCOMMS_A CCOUNT_AP_M AP	Activity	PegaCM-CFF	08-04-02	PegaComm-

Name	Rule type	Ruleset	Ruleset version	Applies to
AmountWithDollar	Control	PegaCM-CFF	08-04-01	
OpenCOMMS_COMPANY_MASTER	Activity	PegaCM-CFF	08-04-02	PegaComm-
OpenCOMMS_CUSTOMER_ACOUNT_MAP	Activity	PegaCM-CFF	08-04-02	PegaComm-
BrowseCOMM_S_ACCOUNT_AP_MAP	Activity	PegaCM-CFF	08-04-02	PegaComm-
BrowseCOMM_S_ACCOUNT_MASTER	Activity	PegaCM-CFF	08-04-02	PegaComm-
BrowseCOMM_S_COMPANY_MASTER	Activity	PegaCM-CFF	08-04-02	PegaComm-
BrowseCOMM_S_CUSTOMER_ACCOUNT_MP	Activity	PegaCM-CFF	08-04-02	PegaComm-
BrowseCOMM_S_CUSTOMER_ADDRESSES	Activity	PegaCM-CFF	08-04-02	PegaComm-
BrowseCOMM_S_CUSTOMER_MASTER	Activity	PegaCM-CFF	08-04-02	PegaComm-

Name	Rule type	Ruleset	Ruleset version	Applies to
OpenCOMMS_ACCOUNT_AP_MAP	Activity	PegaCM-CFF	08-04-02	PegaComm-
OpenCOMMS_CUSTOMER_MASTER	Activity	PegaCM-CFF	08-04-02	PegaComm-
RetrieveActiveProductList	Activity	PegaCM-CFF	08-04-02	PegaComm-
OpenCOMMS_ALL_CUSTOMERS	Activity	PegaCM-CFF	08-04-02	PegaComm-
ShowImages	Control	PegaCM-CFF	08-04-01	
RetrieveProductList	Activity	PegaCM-CFF	08-04-02	PegaComm-
GoogleMap	Control	PegaCM-CFF	08-04-01	
pyCaseWorkerPortalNav	Section	PegaCM-CFF	08-04-01	Data-Portal
DeleteCOMMS_ACCOUNT_MASTER	Activity	PegaCM-CFF	08-04-02	PegaComm-
DeleteCOMMS_CUSTOMER_MASTER	Activity	PegaCM-CFF	08-04-02	PegaComm-
GetCountryList	Activity	PegaCM-CFF	08-04-02	PegaComm-
GetStatesOfCountry	Activity	PegaCM-CFF	08-04-02	PegaComm-

Name	Rule type	Ruleset	Ruleset version	Applies to
OpenCOMMS_ACCOUNT_MA STER	Activity	PegaCM-CFF	08-04-02	PegaComm-
OpenPRODUC T_TYPE_MASTE R	Activity	PegaCM-CFF	08-04-02	PegaComm-
ParseSQLError	Activity	PegaCM-CFF	08-04-02	PegaComm-

### 8.3 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.3 release.

Name	Ruleset	Ruleset version	Applies to
RelatedEntities	PegaPS	08-03-01	PegaPS-Data-Entity
PostEvidence	PegaPS	08-03-01	PegaPS-Work
RelatedEntities	PegaPS	08-03-01	PegaPS-Data-Entity
SetStatusLabel	PegaPS	08-03-01	PegaPS-Data- Address
EntityHasCases	PegaPS	08-03-01	PegaPS-Data-Entity
ViewAllCases	PegaPS	08-03-01	PegaPS-Work
InvestigationAssoci ationControl	PegaPSICM	08-03-01	
RelatedEntities	PegaPSICM	08-03-01	PegaPS-Data-Entity
RelatedEvents	PegaPSICM	08-03-01	PegaPS-Work
RelatedAllEventsList	PegaPSICM	08-03-01	PegaPS-Work-ICM

Name	Ruleset	Ruleset version	Applies to
SubjectRelatedCasesListAll	PegaPSICM	08-03-01	PegaPS-Work-ICM
RelatedInvestigationsVisual	PegaPSICM	08-03-01	PegaPS-Work
RelatedAllSubjects	PegaPSICM	08-03-01	PegaPS-Work-ICM

## 8.2 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.2 release.

Name	Ruleset	Ruleset version	Applies to
Pgpwebsiteheader	PegaPS	08-02-01	@baseclass
Procurementswebsiteheader	PegaPS	08-02-01	@baseclass
WebLoginComplaints	PegaPS	08-02-01	@baseclass
WebLoginPGP	PegaPS	08-02-01	@baseclass
WebLoginProcurements	PegaPS	08-02-01	@baseclass

## 8.1 Deprecated rules

The following table lists the deprecated rules of Pega Government Platform for 8.1 release.

Name	Ruleset	Ruleset version	Applies to
LoadBusinessContent	PegaPS	08-01-01	PegaPS-Data-Context-Business

Name	Ruleset	Ruleset version	Applies to
LoadClassContext	PegaPS	08-01-01	PegaPS-Data-Context-Class
D_StateList	PegaPSInt	08-01-01	PegaPS-Int-Config-State

## Withdrawn rules

This page lists the list of withdrawn rules of Pega Government Platform for 8.x releases.

### 8.8 Withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.8 release.

Name	Ruleset	Ruleset version	Rule Type
PGP 7.40	PegaPS	08-08-01	Product
webwb Bootstrap css	PegaPS	08-08-01	Text File
webwb Custom css	PegaPS	08-08-01	Text File
webwb ProgressIndicator js	PegaPS	08-08-01	Text File

### 8.7 withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.7 release.

Name	Ruleset	Ruleset version	Applies to
pgpDefaultSecured	PegaPS	08-07-01	
CustomRole	PegaPS	08-07-01	Data-Party

Name	Ruleset	Ruleset version	Applies to
pxDeviceType	PegaPS	08-07-01	Index-CircumstanceDefinition
pxIsMobileDevice	PegaPS	08-07-01	Index-CircumstanceDefinition
ProgressColor	PegaPS	08-07-01	PegaPS
ProgressTextSuffix	PegaPS	08-07-01	PegaPS
DispositionType	PegaPS	08-07-01	PegaPS-Data-Config
DateDecimal	PegaPS	08-07-01	PegaPS-Data-Context-Business
MergedFromKey	PegaPS	08-07-01	PegaPS-Data-Snapshot
SetAppLoginPage	PegaPS	08-07-01	Code-Security
NoticeHeaderBkp	PegaPS	08-07-01	PegaPS-Work
RachGrantConsentAgreement	PegaPS	08-07-01	PegaPS-Work
BusinessBranchDetailsRO1	PegaPS	08-07-01	PegaPS-Data-Entity-Business
IACAuthentication	PegaPS	08-07-01	Code-Security
DeleteDocList	PegaPS	08-07-01	Pega-Ext-ExpExplorer-Configuration-Document
AssociateRelatedEvents	PegaPS	08-07-01	PegaPS-Work-Event-Initiate

Name	Ruleset	Ruleset version	Applies to
ValidateAssociates	PegaPS	08-07-01	PegaPS-Work-Manage-Associate
getEmailSenderInfo	PegaPS	08-07-01	PegaPS-Work-RequestAccess
PostAddAttachments	PegaPS	08-07-01	PegaPS-Data
pyAttachContent	PegaPS	08-07-01	PegaPS-Work
AUTSetAttachmentToFile	PegaPS	08-07-01	PegaPS
AUTSetLinkData	PegaPS	08-07-01	PegaPS-Data-Link
EndUser	PegaPS	08-07-01	Data-Portal
PreReviewDocument	PegaPS	08-07-01	PegaPS-Work
SetRadiusRange	PegaPS	08-07-01	PegaPS-Work
SetRelatedEvents	PegaPS	08-07-01	PegaPS-Work-Event-Initiate
SetLinkedFromTo	PegaPS	08-07-01	PegaPS-Work-Manage-Associate
SetCaseStatusForSLA	PegaPS	08-07-01	PegaPS-Work-TransferOwnership
PegaPSEndUser	PegaPS	08-07-01	Data-Portal
ReviewRequest_3	PegaPS	08-07-01	PegaPS-Work-RequestAccess
IsMultipleConfiguredCounties	PegaPS	08-07-01	@baseclass
IsSingleConfiguredCountry	PegaPS	08-07-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
PGPRulesets_pyruleset	PegaPS	08-07-01	@baseclass
DuplicateTemplate	PegaPS	08-07-01	Pega-Ext-ExpExplorer-Configuration-InterviewTemplate
IsAttachmentTypeDocument	PegaPS	08-07-01	PegaPS
ShowCaseActions	PegaPS	08-07-01	PegaPS-Work
ShowManageTeam	PegaPS	08-07-01	PegaPS-Work
NonMandatoryAddressFormWithCurrentLocation	PegaPS	08-07-01	PegaPS-Data-Address
AnonymousReporting	PegaPS	08-07-01	PegaPS-Data-Event
CaseHistoryTab	PegaPS	08-07-01	PegaPS-Work
EventDetailsPreview	PegaPS	08-07-01	PegaPS-Work-Event
RegistrationInfo	PegaPSUserServices	08-07-01	PegaPS-Work-User-Registration
D_DocumentSourceOptions	PGPCosmos	08-07-01	PegaSocial-Document
InterviewReport	PGPCosmos	08-07-01	PegaPS-Data-Interview
TestTraining	PGPCosmos	08-07-01	PegaPS-Data-Training
ReportPDFWrapper	PGPCosmos	08-07-01	PegaPS-Work

Name	Ruleset	Ruleset version	Applies to
EnterCaseDetails	PGPCosmos	08-07-01	PegaSocial-Document
OpenByHandle	PGPCosmos	08-07-01	@baseclass
IsApplicableForBusiness	PGPCosmos	08-07-01	PegaPS-Work-ApplicationRequest
IsApplicableForFacility	PGPCosmos	08-07-01	PegaPS-Work-ApplicationRequest
IsApplicableForPerson	PGPCosmos	08-07-01	PegaPS-Work-ApplicationRequest
CreateForm_Default	PGPCosmos	08-07-01	PegaPS-Work-Entity-Vehicle-Add
LoadDocumentSourceOptions	PGPCosmos	08-07-01	Code-Pega-List
Apply	PGPCosmos	08-07-01	PegaPS-Work-ApplicationRequest
EditAction	PGPCosmosOverride	08-07-01	PegaPS-Work-ActivityPlan
EditAction	PGPCosmosOverride	08-07-01	PegaPS-Work-Goal
EditAction	PGPCosmosOverride	08-07-01	PegaPS-Work-Task
pxDeviceType	PegaPSICM	08-07-01	Index-CircumstanceDefinition
ApplicationRequest	ICMCosmos	08-07-01	PegaPS-Work-ApplicationRequest
ReviewWorkBasket			
UserPortal	ICMCosmos	08-07-01	PegaPS-Data-Portal

Name	Ruleset	Ruleset version	Applies to
SelectItems	ICMCosmos	08-07-01	PegaPS-Work-ICM
ReportPDFWrapper	ICMCosmos	08-07-01	PegaPS-Work-ICM-Investigation-Initiate
pyPopulateCaseContentsInner	PegaPSICM	08-07-01	Assign-Worklist
pyPopulateCaseContentsInner	ICMCosmos	08-07-01	Assign-Worklist
getEmailSenderInfo	PegaPSICM	08-07-01	PegaPS-Work-ICM-Investigation-Initiate
ICMAgent	PegaPSICMPortals	08-07-01	Data-Portal
CaseWorker	PegaPSICM	08-07-01	Data-Portal

## 8.6 Withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.6 release.

Name	Ruleset	Ruleset version	Applies to
pyEmail	PegaPS	08-06-01	@baseclass
PGP 08.10	PegaPS	08-06-01	
PGPSampleData 7.40	PegaPS	08-06-01	
PGPSampleData 8.3	PegaPS	08-06-01	
PGPSampleData 8.4	PegaPS	08-06-01	
PGP_ICM_Demo 7.40	PegaPS	08-06-01	

Name	Ruleset	Ruleset version	Applies to
gella-pegasystems-com-CodeFragment	PegaPS	08-06-01	
AddAssociates	PegaPS	08-06-01	PegaPS-Work-Manage-Associate
pyCaseContent	PegaPS	08-06-01	PegaPS-Work-Manage-Education
pyCaseContent	PegaPS	08-06-01	PegaPS-Work-Manage-Experience
pyUploadFile	PegaPS	08-06-01	Data-WorkAttach-File
pyCaption CommType	PegaPS	08-06-01	PegaPS-Data-Comm-Phone
AddAssociates	PegaPS	08-06-01	PegaPS-Work-Manage-Associate
Web-Login-Member	PegaPS	08-06-01	@baseclass
Web-Login-test	PegaPS	08-06-01	@baseclass
InitiateEvent	PegaPS	08-06-01	PegaPS-Work-Event-Initiate
InitiateAskQuestion	PegaPS	08-06-01	PegaPS-Work-Question
SetMergeStatus	PegaPS	08-06-01	PegaPS-Data-Entity
PersistInterview	PegaPS	08-06-01	PegaPS-Work
ForgotPassword	PegaPSUserServices	08-06-01	PegaPS-Work-User-ForgotPassword
Registration	PegaPSUserServices	08-06-01	PegaPS-Work-User-Registration
EditAction	PGPCosmos	08-06-01	PegaPS-Work

Name	Ruleset	Ruleset version	Applies to
pyCaption CreateNew	PGPCosmos	08-06-01	PegaPS-Work- Manage
CaseDetails	PGPCosmosOverrid e	08-06-01	PegaPS-Work-Event
pyRefresh	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Business- View
pyWorkCommonAct ions	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Business- View
pyRefresh	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Facility-View
pyWorkCommonAct ions	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Facility-View
pyRefresh	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Household- View
pyWorkCommonAct ions	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Household- View
pyRefresh	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Person-View
pyWorkCommonAct ions	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Person-View
pyRefresh	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Vehicle-View
pyWorkCommonAct ions	PGPCosmosOverrid e	08-06-01	PegaPS-Work- Entity-Vehicle-View

Name	Ruleset	Ruleset version	Applies to
ProgressIndicator	PegaPSICM	08-06-01	
AssociateEvent_RDL	ICMCosmos	08-06-01	PegaPS-Data-Event
SearchCriteria	ICMCosmos	08-06-01	PegaPS-Data-Event
pyCaseMainInner	PegaPSICM	08-06-01	PegaPS-Work- Manage
InitializeInterview	PegaPSICM	08-06-01	PegaPS-Work
IsUnderInvestigation	PegaPSICM	08-06-01	PegaPS-Work-ICM- Investigation
PgpServices	PGPOVERRIDE	08-06-01	
webwb Registration.js	PGPOVERRIDE	08-06-01	
UserServices	PGPOVERRIDE	08-06-01	
CustomAuthentication	PGPOVERRIDE	08-06-01	Code-Security
CustomAuthVerification	PGPOVERRIDE	08-06-01	Code-Security
Web-Login-Member	PGPOVERRIDE	08-06-01	@baseclass
Web-Login-test	PGPOVERRIDE	08-06-01	@baseclass
WebLoginMain	PGPOVERRIDE	08-06-01	@baseclass

## 8.5 Withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.5 release.

Name	Ruleset	Ruleset version	Applies to
Inline	PegaPS	08-05-01	PegaPS

Name	Ruleset	Ruleset version	Applies to
Order	PegaPS	08-05-01	PegaPS
SLAName	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
pyCaption Investigation	PegaPS	08-05-01	@baseclass
wss normalizemin css	PegaPS	08-05-01	
Settings	PegaPS	08-05-01	PegaPS-Landing
AttachmentFeedDe scription	PegaPS	08-05-01	
ProgressIndicator	PegaPS	08-05-01	
pyAttachmentFeed Description	PegaPS	08-05-01	
RangeSlider	PegaPS	08-05-01	
PGFooter	PegaPS	08-05-01	@baseclass
PgpNavigation	PegaPS	08-05-01	@baseclass
PgpServices	PegaPS	08-05-01	@baseclass
Pgpwebsiteheader	PegaPS	08-05-01	@baseclass
ProcurementService s	PegaPS	08-05-01	@baseclass
Procurementswebsi teheader	PegaPS	08-05-01	@baseclass
SliderModalTemplat eStyles	PegaPS	08-05-01	@baseclass
ModalSlider	PegaPS	08-05-01	PegaPS

Name	Ruleset	Ruleset version	Applies to
Are you sure you want to delete from this model?	PegaPS	08-05-01	PegaPS-Data-Address
Are you sure you want to delete from this model?	PegaPS	08-05-01	PegaPS-Data-Association
Are you sure you want to delete from this model?	PegaPS	08-05-01	PegaPS-Data-Education
Are you sure you want to delete from this model?	PegaPS	08-05-01	PegaPS-Data-Entity-Person
Are you sure you want to delete from this model?	PegaPS	08-05-01	PegaPS-Data-Training
DocumentationReport	PegaPS	08-05-01	PegaPS-Work
pyCaseMain	PegaPS	08-05-01	PegaPS-Work
pyB2CCaseMainConfig	PegaPS	08-05-01	PegaPS-Work
EventDetails	PegaPS	08-05-01	PegaPS-Work-Event-Initiate
pyCaseMain	PegaPS	08-05-01	Work-
AssessmentConfiguration	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
AssessmentConfiguration_RG	PegaPS	08-05-01	Rule-PegaQ-Questionnaire

Name	Ruleset	Ruleset version	Applies to
AssessmentList	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
AssessmentList_RG	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
AssessmentNoResults	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
pyMainMenu	PegaPS	08-05-01	Pega-Landing
pyCaption Agent	PegaPS	08-05-01	@baseclass
pyCaption Airport	PegaPS	08-05-01	@baseclass
pyCaption Any	PegaPS	08-05-01	@baseclass
pyCaption High	PegaPS	08-05-01	@baseclass
pyCaption Russian	PegaPS	08-05-01	@baseclass
pyActionPrompt Click to collapse	PegaPS	08-05-01	@baseclass
pyActionPrompt Click to expand	PegaPS	08-05-01	@baseclass
pyButtonLabel Request access	PegaPS	08-05-01	@baseclass
pyCaption Active	PegaPS	08-05-01	@baseclass
pyCaption Administrator	PegaPS	08-05-01	@baseclass
pyCaption Apply	PegaPS	08-05-01	@baseclass
pyCaption Approve	PegaPS	08-05-01	@baseclass
pyCaption Audit	PegaPS	08-05-01	@baseclass
pyCaption Blue	PegaPS	08-05-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
pyCaption CaseManager	PegaPS	08-05-01	@baseclass
pyCaption CaseWorker	PegaPS	08-05-01	@baseclass
pyCaption Client	PegaPS	08-05-01	@baseclass
pyCaption Compliant	PegaPS	08-05-01	@baseclass
pyCaption Critical	PegaPS	08-05-01	@baseclass
pyCaption Current operator	PegaPS	08-05-01	@baseclass
pyCaption Custom	PegaPS	08-05-01	@baseclass
pyCaption Customer	PegaPS	08-05-01	@baseclass
pyCaption days	PegaPS	08-05-01	@baseclass
pyCaption Dependent	PegaPS	08-05-01	@baseclass
pyCaption Description	PegaPS	08-05-01	@baseclass
pyCaption Details	PegaPS	08-05-01	@baseclass
pyCaption Document	PegaPS	08-05-01	@baseclass
pyCaption E-Mail	PegaPS	08-05-01	@baseclass
pyCaption Employer	PegaPS	08-05-01	@baseclass
pyCaption English	PegaPS	08-05-01	@baseclass
pyCaption Entity Type	PegaPS	08-05-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
pyCaption Event	PegaPS	08-05-01	@baseclass
pyCaption Experience	PegaPS	08-05-01	@baseclass
pyCaption Expired	PegaPS	08-05-01	@baseclass
pyCaption Fax	PegaPS	08-05-01	@baseclass
pyCaption First Name	PegaPS	08-05-01	@baseclass
pyCaption French	PegaPS	08-05-01	@baseclass
pyCaption German	PegaPS	08-05-01	@baseclass
pyCaption Green	PegaPS	08-05-01	@baseclass
pyCaption Home	PegaPS	08-05-01	@baseclass
pyCaption Home phone	PegaPS	08-05-01	@baseclass
pyCaption I	PegaPS	08-05-01	@baseclass
pyCaption IN	PegaPS	08-05-01	@baseclass
pyCaption Inactive	PegaPS	08-05-01	@baseclass
pyCaption Individual	PegaPS	08-05-01	@baseclass
pyCaption Last Login	PegaPS	08-05-01	@baseclass
pyCaption Last Name	PegaPS	08-05-01	@baseclass
pyCaption Link	PegaPS	08-05-01	@baseclass
pyCaption List	PegaPS	08-05-01	@baseclass
pyCaption Low	PegaPS	08-05-01	@baseclass
pyCaption Map	PegaPS	08-05-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
pyCaption Price	PegaPS	08-05-01	@baseclass
pyCaption Primary	PegaPS	08-05-01	@baseclass
pyCaption Medium	PegaPS	08-05-01	@baseclass
pyCaption months	PegaPS	08-05-01	@baseclass
pyCaption Navigation	PegaPS	08-05-01	@baseclass
pyCaption Nearby Cases	PegaPS	08-05-01	@baseclass
pyCaption New	PegaPS	08-05-01	@baseclass
pyCaption No	PegaPS	08-05-01	@baseclass
pyCaption No attachments	PegaPS	08-05-01	@baseclass
pyCaption Notes	PegaPS	08-05-01	@baseclass
pyCaption Experience	PegaPS	08-05-01	PegaPS-Data-Entity-Person
pyCaption Last Login	PegaPS	08-05-01	PegaPS-Data-Entity-Person
pyCaption Last Name	PegaPS	08-05-01	PegaPS-Data-Entity-Person
pyCaption Open	PegaPS	08-05-01	@baseclass
pyCaption Operator	PegaPS	08-05-01	@baseclass
pyCaption Other	PegaPS	08-05-01	@baseclass
pyCaption Others	PegaPS	08-05-01	@baseclass
pyCaption Pager	PegaPS	08-05-01	@baseclass
pyCaption Parent	PegaPS	08-05-01	@baseclass
pyCaption Partner	PegaPS	08-05-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
pyCaption Personal	PegaPS	08-05-01	@baseclass
pyCaption Pink	PegaPS	08-05-01	@baseclass
pyCaption Province	PegaPS	08-05-01	@baseclass
pyCaption Question	PegaPS	08-05-01	@baseclass
pyCaption Read-only	PegaPS	08-05-01	@baseclass
pyCaption Reassign	PegaPS	08-05-01	@baseclass
pyCaption Reference	PegaPS	08-05-01	@baseclass
pyCaption Reject	PegaPS	08-05-01	@baseclass
pyCaption Related Cases	PegaPS	08-05-01	@baseclass
pyCaption Secondary	PegaPS	08-05-01	@baseclass
pyCaption Service	PegaPS	08-05-01	@baseclass
pyCaption Services	PegaPS	08-05-01	@baseclass
pyCaption Single	PegaPS	08-05-01	@baseclass
pyCaption Specific user	PegaPS	08-05-01	@baseclass
pyCaption SSN	PegaPS	08-05-01	@baseclass
pyCaption Storage	PegaPS	08-05-01	@baseclass
pyCaption Take Action	PegaPS	08-05-01	@baseclass
pyCaption Target	PegaPS	08-05-01	@baseclass
pyCaption Tenant	PegaPS	08-05-01	@baseclass
pyCaption Timeline	PegaPS	08-05-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
pyCaption Tip	PegaPS	08-05-01	@baseclass
pyCaption Training	PegaPS	08-05-01	@baseclass
pyCaption Update	PegaPS	08-05-01	@baseclass
pyCaption User Name	PegaPS	08-05-01	@baseclass
pyCaption Users	PegaPS	08-05-01	@baseclass
pyCaption Withdraw	PegaPS	08-05-01	@baseclass
pyCaption Work queue	PegaPS	08-05-01	@baseclass
pyCaption Workbasket	PegaPS	08-05-01	@baseclass
pyCaption Yes	PegaPS	08-05-01	@baseclass
WebLoginComplaints	PegaPS	08-05-01	@baseclass
WebLoginPGP	PegaPS	08-05-01	@baseclass
WebLoginProcurements	PegaPS	08-05-01	@baseclass
DeleteEducation	PegaPS	08-05-01	PegaPS-Data-Education
DeleteLocation	PegaPS	08-05-01	PegaPS-Data-Address
DeleteAssociation	PegaPS	08-05-01	PegaPS-Data-Association
AddToDeleteList	PegaPS	08-05-01	PegaPS-Data-Entity-Associate

Name	Ruleset	Ruleset version	Applies to
DeleteRelation	PegaPS	08-05-01	PegaPS-Data-Entity-Person
DeleteTraining	PegaPS	08-05-01	PegaPS-Data-Training
PreAssessmentConfiguration	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
RemoveTempId	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
SetTempIdForNewRow	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
IsPGPApplications	PegaPS	08-05-01	PegaPS-Data-Context-Application
PegaPS_bkp_23Jun	PegaPS	08-05-01	
AssessmentConfigurationExists	PegaPS	08-05-01	Rule-PegaQ-Questionnaire
IsSampleData	PegaPS	08-05-01	Rule-Admin-Product
PegaPSUIKit7_07July_bkp	PegaPS	08-05-01	
LayoutGroup	PegaPSUI	08-05-01	PegaPS
Footer	PegaPSUserServices	08-05-01	PegaPS-Work-User-Registration
Header	PegaPSUserServices	08-05-01	PegaPS-Work-User-Registration
Intake	PGPCosmos	08-05-01	PegaPS-Work
Intake	PGPCosmos	08-05-01	PegaPS-Work
.BuiltOnAppCSV	PegaPSICM	08-05-01	PegaPS

Name	Ruleset	Ruleset version	Applies to
InvestigationAssociationControl	PegaPSICM	08-05-01	
ConfirmDeactivate	PegaPSICM	08-05-01	PegaPS-Data-Config-ActivityPlan
RelatedEntities	PegaPSICM	08-05-01	PegaPS-Data-Entity
RelatedInvestigationsVisual	PegaPSICM	08-05-01	PegaPS-Work
RelatedAllEventsList	PegaPSICM	08-05-01	PegaPS-Work-ICM
SubjectRelatedCasesListAll	PegaPSICM	08-05-01	PegaPS-Work-ICM
RelatedInvestigationsVisual	PegaPSICM	08-05-01	PegaPS-Work
RelatedAllSubjects	PegaPSICM	08-05-01	PegaPS-Work-ICM

## 8.4 Withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.4 release.

Name	Ruleset	Ruleset version	Applies to
ConfirmExternal	PegaPS	08-04-01	PegaPS
ConfirmExternal	PegaPS	08-04-01	PegaPS
Log off	PegaPS	08-04-01	Code-Security
Database-Authorization-NoModifyAccess	PegaPS	08-04-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
Database-Authorization-NoOpenAccess	PegaPS	08-04-01	@baseclass
pyCaption Category	PegaPS	08-04-01	@baseclass
pyCaption Create	PegaPS	08-04-01	@baseclass
pyCaption Create new	PegaPS	08-04-01	@baseclass
pyCaption Email	PegaPS	08-04-01	@baseclass
pyCaption Female	PegaPS	08-04-01	@baseclass
pyCaption First Name	PegaPS	08-04-01	@baseclass
pyCaption Georgia	PegaPS	08-04-01	@baseclass
pyCaption Male	PegaPS	08-04-01	@baseclass
pyCaption My Cases	PegaPS	08-04-01	@baseclass
pyCaption Name	PegaPS	08-04-01	@baseclass
pyCaption Owner	PegaPS	08-04-01	@baseclass
pyCaption Person	PegaPS	08-04-01	@baseclass
pyCaption Phone	PegaPS	08-04-01	@baseclass
pyCaption Role	PegaPS	08-04-01	@baseclass
pyCaption Search	PegaPS	08-04-01	@baseclass
pyCaption Select an image	PegaPS	08-04-01	@baseclass
pyCaption Subject	PegaPS	08-04-01	@baseclass
Web-Session-Return	PegaPS	08-04-01	@baseclass

Name	Ruleset	Ruleset version	Applies to
Web-Session-Return	PegaPS	08-04-01	@baseclass
DisplayAttachmentFile	PegaPS	08-04-01	PegaPS
pySecureFeatures	PegaPS	08-04-01	@baseclass
ShowUploadAvatar	PegaPS	08-04-01	PegaPS-Data-Entity-Person
ShowUploadAvatar	PegaPS	08-04-01	PegaPS-Data-Entity-Vehicle
LocationRO	PegaPSICM	08-04-01	PegaPS-Data-Address
pyCaseDetails	PegaPSICM	08-04-01	PegaPS-Work-Manage
HaveDocumentDeletePrivileges	PegaPSICM	08-04-01	Link-Attachment

### 8.3 Withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.3 release.

Name	Ruleset	Ruleset version	Applies to
PGPSampleData 08.10	PegaPS	08-03-01	
.Name.FullName	PegaPS	08-03-01	PegaPS-Data-Entity-Person
.Name.FullNameWithSalutation	PegaPS	08-03-01	PegaPS-Data-Entity-Person
TreeNavigation7	PegaPS	08-03-01	PegaPS-Work

Name	Ruleset	Ruleset version	Applies to
pyRDLPaginator	PegaPS	08-03-01	PegaPS
Association	PegaPS	08-03-01	PegaPS-Data-Entity
AssociationList	PegaPS	08-03-01	PegaPS-Data-Entity
ViewCasesAction	PegaPS	08-03-01	PegaPS-Data-Entity
AUTSetDataForManageMembersNewHousehold	PegaPS	08-03-01	PegaPS
AUTSetExistingHHMembersListData	PegaPS	08-03-01	PegaPS
AUTSetDataForManageMembersUseCase2	PegaPS	08-03-01	PegaPS-Work
AUTSetDataForManageMembersUseCase3	PegaPS	08-03-01	PegaPS-Work
AUTSetDataForManageMembersUseCase4	PegaPS	08-03-01	PegaPS-Work
HasAssociations	PegaPS	08-03-01	PegaPS-Data-Entity
ManageAssociates	PegaPSICM	08-03-01	PegaPS-Work-Manage-Associate
Association	PegaPSICM	08-03-01	PegaPS-Data-Entity
MyWorkCasesList	PegaPSICM	08-03-01	PegaPS-Data-Entity
Interview	PegaPSICM	08-03-01	PegaPS-Work
pyCaseAssetsMobile	PegaPSICM	08-03-01	PegaPS-Work
CustomAuditList	PegaPSICM	08-03-01	PegaPS-Work-ICM

Name	Ruleset	Ruleset version	Applies to
CustomAuditModalWindow	PegaPSICM	08-03-01	PegaPS-Work-ICM
CustomAudit	PegaPSICM	08-03-01	PegaPS-Work-ICM
pyRDLPaginator	PegaPSICM	08-03-01	PegaPS-Work-ICM
Subject	PegaPSICM	08-03-01	PegaPS-Work-ICM-Investigation
pyCaseInformation	PegaPSICM	08-03-01	PegaPS-Work-ICM-Investigation-Initiate
CheckDuplicateAssociation	PegaPSICM	08-03-01	PegaPS-Work-Manage-Associate
ShowInWork	PegaPSICM	08-03-01	PegaPS
AddNote	PegaPSICM	08-03-01	PegaPS-Work-ICM
CustomAudit	PegaPSICM	08-03-01	PegaPS-Work-ICM
pyDefault	PegaPSICM	08-03-01	PegaPS-Work
InitializeInterview_Ext	PegaPSICM	08-03-01	PegaPS-Work-ICM
InitializeEvidence_Ext	PegaPSICM	08-03-01	PegaPS-Work-ICM-Investigation
PostEvidence_Ext	PegaPSICM	08-03-01	PegaPS-Work-ICM-Investigation
HasPrivilegeToManageAssociates	PegaPSICM	08-03-01	PegaPS
IsOperatorAuthorized	PegaPSICM	08-03-01	PegaPS
IsInvestigation	PegaPSICM	08-03-01	PegaPS-Work-ICM

Name	Ruleset	Ruleset version	Applies to
IsSubject	PegaPSICM	08-03-01	PegaPS-Work-ICM

## 8.2 Withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.2 release.

Name	Ruleset	Ruleset version	Applies to
Entity	PegaPS	08-02-01	PegaPS-Work
pyGridPaginator	PegaPS	08-02-01	Data-
AddNote	PegaPS	08-02-01	PegaPS
Notes	PegaPS	08-02-01	PegaPS
Notes	PegaPS	08-02-01	PegaPS
Actions	PegaPS	08-02-01	PegaPS-Data-Entity-Facility
Actions	PegaPS	08-02-01	PegaPS-Data-Entity-Household
Actions	PegaPS	08-02-01	PegaPS-Data-Entity-Vehicle
pyCaseHeaderTitle	PegaPS	08-02-01	Work-
SaveNotes	PegaPS	08-02-01	PegaPS
EditInterview	PegaPS	08-02-01	PegaPS-Data-Interview
AddNote	PegaPS	08-02-01	PegaPS
EditEvidence	PegaPS	08-02-01	PegaPS-Data-Evidence
EditEvidence	PegaPS	08-02-01	PegaPS-Work

Name	Ruleset	Ruleset version	Applies to
EditInterview	PegaPS	08-02-01	PegaPS-Work
PostAddNote	PegaPS	08-02-01	PegaPS
PreAddNote	PegaPS	08-02-01	PegaPS
PostEditInterview	PegaPS	08-02-01	PegaPS-Data-Interview
PreEditEvidence	PegaPS	08-02-01	PegaPS-Work
PostEditEvidence	PegaPS	08-02-01	PegaPS-Work
PostEditInterview	PegaPS	08-02-01	PegaPS-Work
SetInterviewTitle	PegaPS	08-02-01	PegaPS-Work
Notes	PegaPSICM	08-02-01	PegaPS
NotesHeader	PegaPSICM	08-02-01	PegaPS
ViewNotes	PegaPSICM	08-02-01	PegaPS
SaveEvidenceAssociations	PegaPSICM	08-02-01	PegaPS-Work
PersistEmbeddedPage	PegaPSICM	08-02-01	PegaPS-Data-Interview
PersistInterview	PegaPSICM	08-02-01	PegaPS-Work
AddEvidence	PegaPSICM	08-02-01	PegaPS-Work
EditEvidence	PegaPSICM	08-02-01	PegaPS-Work
SetInterviewID	PegaPSICM	08-02-01	PegaPS-Data-Interview
InitializeInterview	PegaPSICM	08-02-01	PegaPS-Work
SetInterviewTitle	PegaPSICM	08-02-01	PegaPS-Work
InitializeInterview	PegaPSICM	08-02-01	PegaPS-Work-ICM

## 8.1 Withdrawn rules

The following table lists the withdrawn rules of Pega Government Platform for 8.1 release.

Name	Ruleset	Ruleset version	Applies to
ShowAttachmentList	PegaPS	08-01-01	PegaPS-Data-Attachment-File
GenerateVerificationCode	PegaPSUserServices	08-01-01	PegaPS-Work-User-Registration

## Patch

This book describes the changes for Pega Government Platform (PGP) patch releases. For information about changes in major and minor releases, select the appropriate product version on the [Pega Government Platform](#) product pages and refer to the release notes.

**Note:**

These patch release notes include all issues resolved across patch releases. For each issue, a reference number is provided, and the prefix of the reference number indicates the issue type. You can use the reference number of an issue in your related conversations with [Pega Support](#).

 The release notes include the following issue types:

**INCs**

Customer-reported incidents. For example, INC-183895.

**SRs**

Support requests, which were used instead of incidents in older patches. For example, SR-D79601.

**ISSUEs**

Pega-identified issues. They might or might not be related to customer-reported incidents. For example, ISSUE 654263 (which might also be written as just 654263).

**SEs**

Sustenance engineering activities. For example, SE-60265.

Starting Q2 2021, all customer-reported issues are logged as INCs. You can view INCs that you logged in the [My Support Portal](#). INCs logged by other Pega customers, and all other issue types (SR, ISSUE, and SE), are available in Pega internal tracking systems, in addition to these patch release notes.

- [8.8.1 patch release](#)

## 8.8.1 patch release

This section provides information about changes made in Pega Government Platform 8.8.1 patch release.

- Minor changes made to application labels.
- Changes to application rules to include only applications built on Theme Cosmos in the new application wizard.

**Note:**

- ⓘ You can click on **Search all types** option from the new application wizard to access the applications built on UI Kit.

# Install

- [Completing the prerequisite tasks](#)
- [Backing up your system](#)
- [Installing the application](#)

## Completing the prerequisite tasks

Before you installupdate your application, ensure that you complete all of the following tasks.

1. Consult with your database administrator to determine whether the install process can make automatic changes to the database schema or if the database administrator will manually apply the DDL files that include schema changes.
2. Install the latest version of Pega Platform 8.8, and ensure that you can log in as an administrator. For more information, see the *Pega Platform Install Guide* for your environment on the [Deploy Pega Platform](#) page.

Patch releases are available through Pega's standard software delivery process. You can request software through Pega Software Distribution or by contacting GCS.

3. Apply any required hotfixes by using the Hotfix Manager. For more information, see [Applying hotfixes](#). To review the Pega Government Platform hotfixes without using the Hotfix Manager, see the Pega Government Platform [Hotfix](#) page.

## Backing up your system

When installing or updating an application, back up your system after each step to ensure that you can revert to the last working version of the system if you encounter an issue.



**Note:** The deployment process modifies both the data schema and the rules schema. Use a backup procedure that preserves both schemas.

1. Verify that all rules are checked in.
2. Shut down the Pega Platform™ application server.
3. Use your database utilities to complete an offline backup of the Pega database.
4. Back up the configuration and environment files.

If you edited any of the following Pega Platform configuration files in the APP-INF\classes directory of an EAR deployment, or the WEB-INF\classes directory of a WAR deployment, include these files in the backup:

- prbootstrap.properties
- prconfig.xml
- logging file: prlogging.xml or prlog4j2.xml
- web.xml
- pegasrules.keyring or any other .keyring files

5. Back up any third-party or custom JAR files that you installed.

Redeploying the Pega Platform applications might delete these files from your application server.

## Installing the application

To install Pega Government Platform, import the application file, and then complete additional procedures in this guide.

### Before you begin:

Ensure that you complete the prerequisites for this application installation.

### Importing the modularized application file

The system data and rulesets for Pega Government Platform are loaded during this task.

1. Log in to (<https://<hostname>:<port>/prweb>) by entering the administrator ID, for example, administrator@pega.com and the password that you used during installation.
2. In the header of Dev Studio, click Configure > Application > Distribution > Import.
3. Select the **/Rules/PegaGP.jar** file from your distribution media, and then follow the wizard instructions.
4. For schema changes, depending on your site's requirements, select either Automatic or Manual, and then continue following the wizard instructions. If you select Manual, see [Viewing and applying schema changes](#).
5. On the Aged Updates screen, select the Import aged updates check box, and then click Next.
6. When the import is complete, click Done.
7. Apply the required hotfixes by using Hotfix Manager. For more information about applying hotfixes, see [Applying the latest on-premises patch](#).
8. **Optional:** Verify the application rules.
  - a. In the navigation pane of Dev Studio, click Records > Application Definition > Application.
  - b. Confirm that the following application rule is displayed in the list:
    - *PegaGPCosmos 8*
    - *PGPICMCosmos 8*
    - *TimeMgmt 8*
    - *PegaGP 8*
    - *PGPICM 8*
    - *PGPContracts 8*
    - *PGPComplaints 8*

#### What to do next:

Optionally, import the sample data.

## Creating a system administrator operator

Create a system administrator operator for Pega Government Platform by completing the following steps:

1. Log in to (<https://<hostname>:<port>/prweb>) by entering the administrator ID, for example, administrator@pega.com and the password that you used during installation.
2. In the header of Dev Studio, click Create > Organization > Operator ID.
3. Enter a short description, for example, pgpsysadmin, and then click Create and open.
4. In the Application Access section, enter the access group PegaGPCosmos : Administrators.
5. Click the Work tab. Under Organizational unit, click Update.
6. Click PGPGov > Government > Operations, and then click Submit.
7. On the Security tab, click Update password, enter a new password, and then click Submit.
8. Click Save to save the rule.

## Creating operator accounts

If DWA (Directed Web Access) and Listener functionality are used in your application, make sure that the following operators are available in the environment.

OperatorID	Access Group	Team	Starting activity
ICMEventListener	PGPICMCosmos:IC MIntakeSpecialist	IntakeSpecialist	Data- Portal.ShowDeskt o p
PGPEventListener	PegaGPCosmos:Cas eWorker	default@PegaPS	Data- Portal.ShowDeskt o p

<b>OperatorID</b>	<b>Access Group</b>	<b>Team</b>	<b>Starting activity</b>
RequestAccessEmailApprover	PGPICMCosmos:Administrator	default@PegaPS	Data-Portal.ShowDesktop
ICMInvestigationListener	PGPICMCosmos:Administrator	default@PegaPS	Data-Portal.ShowDesktop

For UI Kit apps: For applications built on UI Kit.

<b>OperatorID</b>	<b>Access Group</b>	<b>Work Group</b>	<b>Starting activity</b>
ExternalUserPGP	PegaGP:Caseworker	Default	Data-Portal.ShowDesktop
ExternalUserComplaint	PegaGP:ComplaintsWorker	Default	Data-Portal.ShowDesktop
ExternalUserICM	PegaGP:ICMAgent	Default	Data-Portal.ShowDesktop
ExternalUserProc	PegaGP:ProcurementsWorker	Default	Data-Portal.ShowDesktop
ICMInvestigationListener	PegaGP:ICMAdmin	default@PegaPS	Data-Portal.ShowDesktop

If operators are not available, follow these steps to create operators from the above tables in your Pega Government Platform application.

1. Log in to Pega Platform by using the administrator credentials, and switch to the respective applications and create the corresponding operators listed above.
2. In the header of Dev Studio, click Create > Organization > Operator ID.
3. Enter the Operator ID from the corresponding column in the Operators table and click Create and open.
4. Add the Access group from the corresponding column in the Operators table.

5. Click the Work tab. Under Organizational unit, click Update.
6. Click PGPGov > Government > Operations, and then click Submit.
7. Add the Work group from the corresponding column in the Operators table.
8. On the Security tab, add the starting activity to execute from the corresponding column in the Operators table.
9. Click Update password, enter a new password, and then click Submit.
10. Click Save to save the rule.

## Updating event listeners

Pega Government Platform has a set of event listeners.

To update these listeners with requestor login credentials, complete the following steps.

1. In the header of Dev Studio, click Configure > Integration > Email > Email Listeners.
2. Click the *PGPEventEmailListener* listener name.
3. On the Email listener page, navigate to the Requestor login section and update the password of *PGPEventListener*, which was created in [Creating operator accounts](#).
4. Save the record.
5. Repeat the previous steps for the following event listeners:
  - ICMEventEmailListener
  - RequestAccessApproval
  - InvestigationAcceptEmailListener

## Optional: Importing the sample data

The distribution media includes sample data to help you explore the application capabilities. The sample data includes disabled sample operators that you can enable to learn about the workspaces for each operator role. Importing the sample data is optional, and the sample data that is provided might not align with your business needs.

**Before you begin:**

Ensure that you have already imported the application file.

To import the sample data for , complete the following steps.

1. In the header of Dev Studio, click Configure > Application > Distribution > Import.
2. Select the file from your distribution media.
3. Follow the wizard instructions to import the sample data.

**What to do next:**

Enable the sample operators that you will use to explore the different workspaces. For more information, see [Optional: Enabling sample operator accounts](#).

## Optional: Enabling sample operator accounts

If you imported the sample data, operators are included that allow you to access the sample data. For security purposes, sample operators are disabled by default.

To enable the sample operators, complete the following steps.

1. In the header of Dev Studio, click Configure > Org & Security > Authentication > Operator Access.
2. In the Disabled operators section, select the check box next to the operator ID to enable.
3. Click Enable selected.  
The **Enable Operator** dialog is displayed.
4. Click Submit to confirm that you want to enable the selected operator ID.
5. Click OK to close the dialog box.

**Note:** If you have the john.johnson@uplusfs.com operator in the system and you want to start or continue using the operator, you should change its access group to *CLMFSCIB\_RM*.

## Optional: Installing the sample application

Pega Government Platform includes a sample application with an ICM Demo and PGP Demo.

To install the sample application, complete the following steps:

1. In the header of Dev Studio, click Configure > Application > Distribution > Import.
2. Select the /OptionalComponents/PegaGP\_Sample\_App.jar file from your distribution media, and then follow the wizard instructions.
3. When the import is complete, click Done.

# Update and hotfixes

- [Update](#)
- [Hotfixes](#)

## Update

- [Completing the prerequisite tasks](#)
- [Backing up your system](#)
- [Updating the application](#)

### Completing the prerequisite tasks

Before you installupdate your application, ensure that you complete all of the following tasks.

1. Consult with your database administrator to determine whether the update process can make automatic changes to the database schema or if the database administrator will manually apply the DDL files that include schema changes.
2. Update to the latest version of Pega Platform 8.8, and ensure that you can log in as an administrator. For more information, see the *Pega Platform Update Guide* for your environment on the [Deploy Pega Platform](#) page.

Patch releases are available through Pega's standard software delivery process. You can request software through Pega Software Distribution or by contacting GCS.

3. Apply any required hotfixes by using the Hotfix Manager. For more information, see [Applying hotfixes](#). To review the Pega Government Platform hotfixes without using the Hotfix Manager, see the Pega Government Platform [Hotfix](#) page.

## Backing up your system

When installing or updating an application, back up your system after each step to ensure that you can revert to the last working version of the system if you encounter an issue.

**Important:** Importing the new version of the application can require the execution of column and declare-index population jobs. These jobs run in the background, populating the new columns and declare-indexes that are imported with the application, which requires the update of a large number of records. In PostgreSQL installations, this large-scale update of records can require additional temporary disk space, so ensure that there is enough disk space available for the database to expand accordingly.

- ⚠️ Imported with the application, which requires the update of a large number of records. In PostgreSQL installations, this large-scale update of records can require additional temporary disk space, so ensure that there is enough disk space available for the database to expand accordingly.

- ⓘ **Note:** The deployment process modifies both the data schema and the rules schema. Use a backup procedure that preserves both schemas.

1. Verify that all rules are checked in.
2. Shut down the Pega Platform™ application server.
3. Use your database utilities to complete an offline backup of the Pega database.
4. Back up the configuration and environment files.

If you edited any of the following Pega Platform configuration files in the APP-INF\classes directory of an EAR deployment, or the WEB-INF\classes directory of a WAR deployment, include these files in the backup:

- prbootstrap.properties
- prconfig.xml
- logging file: prlogging.xml or prlog4j2.xml
- web.xml
- pegasrules.keyring or any other .keyring files

5. Back up any third-party or custom JAR files that you installed.

Redeploying the Pega Platform applications might delete these files from your application server.

## Updating the application

To update Pega Government Platform, import the application file, and then complete additional procedures in this guide.

### Before you begin:

Ensure that you complete the prerequisites for this application update.

## Importing the modularized application file

The system data and rulesets for Pega Government Platform are loaded during this task.

**Note:** If you are upgrading from 7.22 or later, to avoid potential import errors,

- ⓘ search for the `safs_work_p2p` table and rename the `ISEXTERNALSYSTEMOFRECORD` column before starting the import.

1. Log in to (`https://<hostname>:<port>/prweb`) by entering the administrator ID, for example, `administrator@pega.com` and the password that you used during installation.
2. In the header of Dev Studio, click **Configure > Application > Distribution > Import**.
3. Select the **/Rules/PegaGP.jar** file from your distribution media, and then follow the wizard instructions.
4. For schema changes, depending on your site's requirements, select either **Automatic** or **Manual**, and then continue following the wizard instructions.  
If you select **Manual**, see [Viewing and applying schema changes](#).
5. On the Aged Updates screen, select the **Import aged updates** check box, and then click **Next**.

6. When the import is complete, click Done.
7. Apply the required hotfixes by using Hotfix Manager. For more information about applying hotfixes, see [Applying the latest on-premises patch](#).
8. If you are using the CSEndUser skin for your existing application, update the application to use the CSEndUser skin.
9. **Optional:** Verify the application rules.
  - a. In the navigation pane of Dev Studio, click Records > Application Definition > Application.
  - b. Confirm that the following application rule is displayed in the list:
    - *PegaGPCosmos 8*
    - *PGPICMCosmos 8*
    - *TimeMgmt 8*
    - *PegaGP 8*
    - *PGPICM 8*
    - *PGPContracts 8*
    - *PGPComplaints 8*

#### What to do next:

Complete the additional post-update tasks described in this guide, if applicable.

## Creating operator accounts

If DWA (Directed Web Access) and Listener functionality are used in your application, make sure that the following operators are available in the environment.

For UI Kit apps: For applications built on UI Kit.

OperatorID	Access Group	Work Group	Starting activity
ExternalUserP GP	PegaGP:Caseworker	Default	Data-Portal.ShowDesktop

OperatorID	Access Group	Work Group	Starting activity
ExternalUserComplaint	PegaGP:ComplaintsWorker	Default	Data-Portal.ShowDesktop
ExternalUserICM	PegaGP:ICMAgent	Default	Data-Portal.ShowDesktop
ExternalUserProc	PegaGP:ProcurementsWorker	Default	Data-Portal.ShowDesktop
ICMInvestigationListener	PegaGP:ICMAdmin	default@PegaPS	Data-Portal.ShowDesktop

For Theme Cosmos apps: For applications built on Theme Cosmos.

OperatorID	Access Group	Work Group	Starting activity
ICMInvestigationListener	PGPICMTheme-Cosmos:Administrator	default@PegaPS	Data-Portal.ShowDesktop

If operators are not available, follow these steps to create operators from the above tables in your Pega Government Platform application.

1. Log in to Pega Platform by using the administrator credentials, and switch to the respective applications and create the corresponding operators listed above.
2. In the header of Dev Studio, click Create > Organization > Operator ID.
3. Enter the Operator ID from the corresponding column in the Operators table and click Create and open.
4. Add the Access group from the corresponding column in the Operators table.
5. Click the Work tab. Under Organizational unit, click Update.
6. Click PGPGov > Government > Operations, and then click Submit.
7. Add the Work group from the corresponding column in the Operators table.
8. On the Security tab, add the starting activity to execute from the corresponding column in the Operators table.
9. Click Update password, enter a new password, and then click Submit.

10. Click Save to save the rule.

## Optional: Importing the sample data

The distribution media includes sample data to help you explore the application capabilities. The sample data includes disabled sample operators that you can enable to learn about the workspaces for each operator role. Importing the sample data is optional, and the sample data that is provided might not align with your business needs.

### Before you begin:

Ensure that you have already imported the application file.

To import the sample data for , complete the following steps.

1. In the header of Dev Studio, click Configure > Application > Distribution > Import.
2. Select the file from your distribution media.
3. Follow the wizard instructions to import the sample data.

### What to do next:

Enable the sample operators that you will use to explore the different workspaces. For more information, see [Optional: Enabling sample operator accounts](#).

## Updating data from the integration layer to the data layer

The Pega Government Platform integration layer has been refactored in the 8.3 release. All tables are now mapped to their respective classes in the data layer. Refactoring activities are provided for each of the applications that are shipped and for each class that is refactored.

To ensure a successful update of your application from Pega Government Platform 8.2 and earlier, review the topics in this section.

## Optional: Enabling backward compatibility with the integration layer

By default, Pega Government Platform 8.3 only works with the data layer. For backward compatibility purposes, you can update two dynamic system settings to enable Pega Government Platform to point back to the integration layer.



**CAUTION:** Do not use the integration layer with data tables as a source.

- *PGP/MasterDataIntegrationRefactor* – to point the Pega Government Platform master data, such as AddressTypes, CommTypes, and so on to the integration layer, change this value to `false`
- *PGP/IntegrationRefactor* – to point the Pega Government Platform non-master data, such as entities, addresses, and so on to the integration layer, change this value to `false`

## Refactoring wrapper activities

For backward compatibility, Pega Government Platform provides an option to resave instances of integration classes as instances of data classes. To perform this refactoring, a few wrapping activities are available. The following table lists the refactoring wrapper activities that are provided for each application. Each activity invokes the refactoring activities of specific classes.



**Note:** Before executing the activities, be sure that you have admin operators for each application pointing to the access group that is specified in the table.

To run the activities, launch the application that is specified in the table and access the respective rulesets from the production rulesets of the access group.

<b>Application</b>	<b>Activity</b>	<b>Ruleset</b>	<b>Access group</b>	<b>Description</b>
PGPI CM	PegaPS.Refactor_IntToData	<i>PegaPSRefactoring:08-03-01</i>	PegaGP:ICMA dmin	Refactors the Int classes for the PGP and ICM applications
PGPI CM	PegaPS.RefactorMasterInfo_IntToData	<i>PegaPSRefactoring:08-03-01</i>	PegaGP:ICMA dmin	Refactors the master Int classes for the PGP and ICM applications
PGPC complaints	PegaPS.RefactorMasterInfo_IntToData	<i>PegaPSComplaintsRefactoring:08-03-01</i>	PegaGP:Com plaintsAdmin	Refactors the master Int classes for the Complaints application

Application	Activity	Ruleset	Access group	Description
PGPContracts	PegaPS.RefactorMasterProclInfo_IntToData	<i>PegaPSProcurementsRefactoring:08-03-01</i>	PegaGP:ProcurementsAdmin	Refactors the master Int classes for the Procurement application

After running the wrapper activities, clear all the node level data pages in all the applications.

## Adding properties to the integration and data layers

When adding properties to the integration and data layers in the Pega Government Platform implementation layer, before executing the activities listed in [Refactoring wrapper activities](#), update the mappings in the *Refactor\_CopyIntToData* data transform, which is specific to the respective data class.

## Customizing report definitions in the implementation layer

When a report definition from an integration class is specialized into the Pega Government Platform implementation layer, perform the same changes to the corresponding report definition of the data layer class.

**For example:**

For example: if any changes are made to the PersonList report definition (*PegaPS-Int-Entity-Person*), perform the same changes to the PersonList (*PegaPS-Data-Entity-Person*) report definition in the implementation layer-specific ruleset.

## Customizing data pages in the implementation layer

When data pages are customized in the Pega Government Platform implementation layer, update the same data page with another data source that retrieves the data from the data layer with data layer-specific report definitions and response data transforms, if required. Use the `IsIntegrationRefactored` when rule to retrieve data from the data layer in the data page.



**Note:** To find the report definition and response data transforms, refer to the data page of the latest Pega Government Platform version.

## Customizing data page response data transforms in the implementation layer

When data page response data transforms are customized in the Pega Government Platform implementation layer, make the same changes to the new response data transforms only if required.



**Note:** To find the new response mapping data transforms for the data pages, refer to the latest Pega Government Platform version data pages.

**For example:**

For example: if the response data transform *CopyPersonIntToData* is customized for the *D\_PersonList* data page, perform the same steps in the new data transform *FetchPersonListDetails*.

## Adding classes to communications and identifiers in the implementation layer

When classes are added to the communications and identifiers wrapper classes in the Pega Government Platform implementation layer, customize the *Refactor\_Identifier* and *Refactor\_Comm* refactoring activities accordingly to meet your requirements.

**For example:**

For example: if a new class *PegaPS-Data-Comm-Twitter* is added for capturing Twitter account details, customize the code of the *Refactor\_Comm* activity to set the respective data class names to the instances of the Twitter class.

## Modifying the State drop-down menu source

If data refactoring is performed by following the preceding sections, modify the source of the State drop-down menu.

For the sections that are overridden that use the StateList report definition (*PegaPS-Int-Config-State*), replace the source by using the *D\_StatesList* data page.

## Modifying the Country drop-down menu source

In Pega Government Platform 8.3, the Country drop-down menu source has been changed from a report definition to a data page.

To use the Pega Government Platform out-of-the-box address-related sections, customize the corresponding data page in the implementation layer.

## State label configuration

In Pega Government Platform 8.3, state labels specific to each country have been moved from the StateLabel decision table (*PegaPS-Data-Address*) to the Country data type.

After upgrading your application, any customizations to the decision table in the implementation layer must be also done in the Country data type. Customizations can be performed by using the Internationalization menu, which is accessed from the Pega Government Platform configuration in App Studio.

## App Studio compliance changes

The Pega Government Platform has been refactored to be in compliant with App Studio in 8.5 release. As part of that, sections that are included in a wrapper section or harness or flow actions with the user level page context have been modified to use the editable data page.

To ensure a successful update of your application from previous releases, Revisit the following rule types that are overridden in implementation layers and which uses the user level pages and make the necessary changes by referring to the corresponding latest rule in the platform.

- Section
- Data transform
- Harness
- Activity

## Extending new case types in the implementation layer

In 8.7 release, PGP includes the Application request, Product and License case types. In 8.8 release, PGP includes the Time entry and Item related Add, Update, and View case

types. To include this case in an implementation application that was built prior, complete the following steps:

1. Create a class record by using the format: <OrgName>-<AppName>-Work-ApplicationRequest, <OrgName>-<AppName>-Work-Program, <OrgName>-<AppName>-Work-Program-License, <OrgName>-<AppName>-Work-TimeEntry, <OrgName>-<AppName>-Work- Entity-Item-Add, <OrgName>-<AppName>-Work-Entity-Item-Update, <OrgName>-<AppName>-Work- Entity-Item-View. For the class group, use the format: <OrgName>-<AppName>-Work.
2. For the directed parent class, based on the case type, use the format: for Application request entities - *PegaPS-Work-ApplicationRequest*. Similarly, use *PegaPS-Work-Program*, *PegaPS-Work-Program-License*, *PegaPS-Work-TimeEntry*, *PegaPS-Work-Entity-Item-Add*, *PegaPS-Work-Entity-Item-Update*, and *PegaPS-Work-Entity-Item-View* and select the Find by name first (Pattern) box.
3. Optimize the properties for the newly created classes that are present in the external mappings of their corresponding directed parent class.
4. Save the class rule forms.

## **Modifying the class context data transform rule**

In Pega Government Platform 8.8, in the *LoadClassContext* data transform (*PegaPS-Data-Context- Application*) (*PegaGPCosmos:08-08-01*), in steps 67-70, new work classes are added.

In Pega Government Platform 8.8, *LoadClassContext* data transform (*PegaPS-Data-Context- Application*) (*PegaGPCosmos:08-08-01*), in steps 149-150, new data classes are added.

Add these steps to your implementation layer-specific data transform, if present.

## **Modifying the business context data transform rule**

In Pega Government Platform 8.8, in the *LoadBusinessContext* data transform, (*PegaPS- Data- Context-Application*) (*PGPCosmos:08-08-01*), new steps: from steps 100 to 108 are introduced.

## Enabling operator creation in user registration functionality

For security reasons, Pega Platform operator creation is disabled in Pega Government Platform 8.4 and later. To enable the creation of an operator, save the **CreatePGPOperator** application setting with owning ruleset **PegaPS** to the implementation application ruleset, and set the value to **True**.

## Optimized properties

A few properties are optimized in 8.6 for the *PegaPS-Work-Event-Initiate*, *PegaPS-Work-ICM-Subject-Initiate*, and *PegaPS-Work-ICM-Investigation-Initiate* classes. In PGP 8.8, few properties are optimized for *PegaPS-Work-Program*, *PegaPS-Work-Program-License* classes.

Follow the same for the implementation layer specific classes to avoid issues in running the report definitions.

## Elasticsearch

In Pega Government Platform 8.6, Elasticsearch is supported for improving the performance of the application for Person, Business, Vehicle, Facility, and Entity data classes and Investigation-Initiate work classes.

The existing list type data pages like D\_PersonList, D\_BusinessList, and so on of the corresponding classes are added with a new data source to fetch the details using Elasticsearch, and this is handled by a single DSS *PGP/ElasticSearch* which is by default set to *False*. Change it to *True* to enable Elasticsearch.

Alternatively, in App Studio go to PGP > Application to change the settings.

You can configure the number of results that Elasticsearch returns by updating the **PGP/DefaultElasticResultCount** DSS, which is by default set to 100 results.

## Performance mode

In 8.6, a new DSS **PGP/PerformanceMode** is introduced to improve the performance of loading the entity pages.

When this DSS is enabled to true (by default its set to false), the entity data pages use Report definition instead of lookup to open the object.

Now entity data pages like D\_Person, D\_Business, and so on use data transform as the source and the corresponding D\_PersonLookup and D\_BusinessLookup has multiple sources defined, based on the setting which sets the data back to D\_Person and D\_Business.

## Event life cycle changes in PGP

In 8.6, a couple of stages are introduced to support the latest functionality. You must make the corresponding changes in the implementation layer-specific Event case, based on your business scenarios.

## IACAuthentication service

Pega included IACAuthentication till the Pega Platform 8.6 release. Pega Platform 8.7 release does not include IACAuthentication, and the recommended approach is to use web mashup. This effects user registration related features like Registration, and the Forgot Password features of PGP Ui-Kit application. Implementation teams must make necessary changes to adopt to the latest Pega Platform 8.7 release changes.

## IACAuthentication activity

In Pega Government Platform 8.7, the IACAuthentication activity is withdrawn from *PegaPS* ruleset. The IACAuthentication activity is moved to the UI Kit specific *PGPOVERRIDE* ruleset, because the PGP application does not use the IACAuthentication activity. Create a copy of this in your implementation ruleset if its being used in implementation application built on Pega Government Platform application. In Pega Government Platform 8.8, to enhance security, step 6 is modified in the

IACAuthentication activity to check for password validation all the time. For the functionality to work, pass the password as parameter in the UserServices HTML Fragment rule along with the existing app name parameter.

For enhancing security, a new step 6 is added in the IACAuthentication activity to check for password validation. To skip this step, save the *CreatePGPOperator* application setting with the owning ruleset PegaPS, to the implementation application ruleset, and set the value to *True*.

## Updates to the US State codes

In Pega Government Platform 8.7, the following state codes are corrected. Implementation teams need to make sure of backward compatibility, while upgrading, if these are being used as-is from the PGP application.

State	Old value	New value
Colorado	CL	CO
Delaware	DL	DE
Maryland	MR	MD

## Changes to entity search data pages in PGP

In Pega Government Platform 8.7, the following search data pages are updated to support AND logical condition along with the existing OR logical condition. The default behavior is updated to AND from OR, because OR condition is applicable only in the search landing pages.

If the search landing pages are updated in implementation layer or the data page is being used elsewhere for OR scenario , update them by passing the value *true* to *UseOROperator* parameter of the data page.

Data pages
D_PersonList

## Data pages

- D\_BusinessList
- D\_FacilityList
- D\_VehicleList

## Updates to case edit actions

In Pega Government Platform 8.7, *Edit case details*, which is a default behavior of Pega Platform, is removed for Transfer ownership, Task, and Request access cases as they are not implemented. To enable the edit action, override the *DisplayEditAction* when condition to the implementation layer with the appropriate condition and add *pyUpdateCaseDetails* flow action to the case type rule. Else, remove the flow action from the case type rule, if present.

## Changes to Registration.js

In PGP 8.7, the *goToLoginPage* and *showHomePage* functions in *Registration.js* file are updated to navigate the URL specific to the invoking application based on the application alias and this is supported by adding a new parameter alias to both the functions.

All the applications consuming these functions as-is need to send the alias parameter to the function where it is being invoked using the *Application.pyProductAlias* property. If the property is empty, set the alias on the application definition rule.

## Changes to Unique ID generation

In Pega Government Platform 8.8, Unique ID generation is changed to improve the performance. To revert to earlier way of generating the Unique ID, Update the Optimize unique ID generation configuration to false.

Follow these steps to set **Optimize unique ID generation** configuration setting to false:

1. In the navigation pane of App Studio, click **Settings > Configurations**.
2. Click the **Settings** icon at the end of **Optimize unique ID generation** row.

Configuration set: PGP						Total 5
Edit plan or goal or task details	if set to true, Allows user to edit plan, goal and task details.	Boolean	true	PGP	6/28/2023	
Entity audit	If set to true, enables the audit for Entities. Make sure the DSS: PGP/PerformanceMode is set to false.	Boolean	false	PGP	7/11/2022	
Item duplicate check	If set to true, enables the duplicate check of items	Boolean	true	PGP	2/25/2022	
Optimize unique id generation	Optimize unique ID generation for data objects	Boolean	true	PGP	3/21/2022	
QP instead of agent	Use queue processor instead of agent	Boolean	true	PGP	7/1/2022	

### PGP Configuration set

3. Select constant value as **False** to revert to earlier way of generating the Unique ID and click **Submit**.

#### Configure setting



Name

Optimize unique id generation

Description

Optimize unique ID generation for data objects

Configuration set

PGP

Configuration scope

Application

Type

Boolean

Default value

Constant

True

> Advanced

Cancel

Submit

### Optimize unique ID generation config setting

## Queue processor for sending emails

In Pega Government Platform 8.8, for case types that use PegaPS agent to initiate sending an email is changed to use Queue processor. To revert to using the agent, update the QP instead of agent configuration to false.

Follow these steps to set the QP instead of agent configuration setting to false:

1. In the navigation pane of App Studio, click Settings > Configurations.
2. Click on Settings icon at the end of the QP instead of agent row.

Configuration set: PGP						Total 5
Edit plan or goal or task details	if set to true, Allows user to edit plan, goal and task details.	Boolean	true	PGP	6/28/2023	 
Entity audit	If set to true, enables the audit for Entities. Make sure the DSS: PGP/PerformanceMode is set to false.	Boolean	false	PGP	7/11/2022	 
Item duplicate check	If set to true, enables the duplicate check of items	Boolean	true	PGP	2/25/2022	 
Optimize unique id generation	Optimize unique ID generation for data objects	Boolean	true	PGP	3/21/2022	 
QP instead of agent	Use queue processor instead of agent	Boolean	true	PGP	7/1/2022	 

*PGP Configuration set*

3. Select constant value as False to revert to agent and click Submit.

**Configure setting** X

Name	QP instead of agent
Description	Use queue processor instead of agent
Configuration set	PGP
Configuration scope	<input type="text" value="Application"/>
Type	Boolean
Default value	<div style="display: flex; align-items: center;"> <span style="border: 1px solid #ccc; padding: 5px; margin-right: 10px;">Constant</span> <span style="border: 1px solid #ccc; padding: 5px; margin-right: 10px;">True</span> <span style="border: 1px solid #ccc; padding: 5px; width: 100px;">▼</span> </div>
<span style="margin-right: 10px;">&gt;</span> Advanced	
<input type="button" value="Cancel"/>	<input style="background-color: #0072BD; color: white; border-radius: 5px; border: none; padding: 5px 10px; font-weight: bold; font-size: 1em; cursor: pointer; width: 150px; height: 30px; margin-left: 10px;" type="button" value="Submit"/>

*Configuration setting*

## Security policies for password validation in user registration

In Pega Government Platform 8.8, an *Edit validate* rule has been added to the password property (CL: PegaPS-Data-Registration) to support security policies for the accounts created by the user registration feature. This has an impact on existing in-flight cases. Take corrective action in the implementation layer, as the new validation is added to the password property, existing accounts with invalid passwords might cause issues or will throw exception. This is applicable only if you are using the user registration feature in PGP.

In addition, the Do not save property data option is enabled for the Confirm password property, which captures the confirm password in text format.

## History and Field audit for entities

In Pega Government Platform 8.8, History and Field audit is enabled for all the supported entities in all PGP application built on Theme Cosmos. To enable this feature after upgrade, follow the below steps:

- Enable the Entity audit configuration from app studio.
- Make sure the DSS PGP/PerformanceMode is set to false and flush the D\_PerformanceMode data page for the DSS changes to take effect.

This feature does not work as expected for auditing list properties like Address List for the entity instances which are created before upgrade.

## Skip Entity persistence in Entity flows

In Pega Government Platform 8.8, to improve performance, the entity persistence logic is skipped when there are no changes performed to the existing details of an entity. This is developed for the below entity related flows in *PegaPS-Work* class.

- BasicPersonalInformation
- PersonInformation
- BusinessInformation
- ManageVehicle
- ManageHousehold
- ManageFacility

## New Declare indexes

In Pega Government Platform 8.8, we have introduced three new declare indexes for supporting the Activity Plan usage feature. The following are the classes to which new declare indexes are added and this is supported only for the cases created from 8.8.

- *PegaPS-Work-Task*
- *PegaPS-Work-Goal*
- *PegaPS-Work-ActivityPlan*

## Changes to VIS.JS

In Pega Government Platform 8.8, we have updated the code in VIS.JS text file to refer to the latest VIS-Network js. As a result, if the code is used in your implementation layer, you must adopt the latest changes of it. In Pega Government Platform, as part of adoption, we have added `shakeTowards: "roots"` as a value to the hierarchical layout settings. Check *EntityAssociationControl* for reference.

The *Vis.JS* file is deprecated and the js is being used directly using the source URL in the control. Check *EntityAssociationControl* for reference and we highly recommend you to follow the same approach as the Vis.js file will be deleted in the upcoming releases.

## Hotfixes

The following tables list required hotfixes the Pega Government Platform application requires. This page also lists hotfixes required for the Pega Foundation for Government application, the Pega application superceded by the Pega Government Platform application.

To request a hotfix, go to [My Support Portal](#). Click New request > For something I need and select Service request > Existing hot fix. Add and verify the hotfix details and click Finish.

Import each type of hotfix in the listed order during the Pega Government Platform or Pega Foundation for Government installation or upgrade:

- Apply **Pega Platform** hotfixes immediately after the Pega Platform installation or upgrade.
- Apply **Pega Government Platform** hotfixes just after you complete the application bundle import.

To see hotfix installation details, see the readme that is included in the hotfix.

- [Hotfixes for Pega Government Platform 8.x](#)

- Hotfixes for Pega Government Platform 7.x

## Hotfixes for Pega Government Platform 8.x

- Pega Government Platform 8.8
- Pega Government Platform 8.7
- Pega Government Platform 8.6
- Pega Government Platform 8.5
- Pega Government Platform 8.4
- Pega Government Platform 8.3
- Pega Government Platform 8.2
- Pega Government Platform 8.1

## Pega Government Platform 8.8

The following table lists the required **Pega Government Platform 8.8** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.8	HFix-85100	<p>BAC pre-registration for activities.</p> <p><b>Note:</b> This hotfix has a prerequisite of Pega Platform patch 8.8.1 and Pega Government</p>

Pega Platform version	Hotfix number	Observed behavior
		<p>Platform patch 8.8.1.</p>

## Pega Government Platform 8.7

The following table lists the required **Pega Government Platform 8.7** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.7	HFix-84901	<p>BAC Pre Registration for activities</p> <p><b>Note:</b> This hotfix has a prerequisite of platform hotfix HFIX-84702.</p>

## Pega Government Platform 8.6

The following table lists the required **Pega Government Platform 8.6** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.6	HFix-84878	<p>BAC Pre Registration for activities</p> <p><b>Note:</b> This hotfix has a prerequisite</p>

Pega Platform version	Hotfix number	Observed behavior
		<p><span style="color: #0070C0;">(i)</span> of platform 8.6.6 patch .</p>
	HFix-82530	<p>Enhancement- App Factory Support for PGP 8.6.</p> <p>Provided Authors Access group for both PegaGPCosmos and PGPICMCosmos apps.</p>
	HFix-80896	Google key configured as default value in parameters.
	HFix-80827	<p>The following issues are addressed:</p> <ul style="list-style-type: none"> <li>• SetObjClass data transform throwing error from @baseclass.</li> <li>• ROI Generation in Cosmos investigation does not pull some new properties.</li> <li>• Work and Data attachments do not prompt correct error in UI Kit.</li> <li>• Support to only RTE and local file for Data</li> </ul>

Pega Platform version	Hotfix number	Observed behavior
		<p>attachments in Cosmos.</p> <ul style="list-style-type: none"> <li>• Removal of default param value (Google key) from the data page parameters.</li> <li>• Subject search issues in Cosmos.</li> <li>• Global context fixes for images.</li> <li>• Merge attribute for business label issue for <i>Ownership type</i> property.</li> </ul>

## Pega Government Platform 8.5

The following table lists the required **Pega Government Platform 8.5** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.5.1	HFix-80897	<ul style="list-style-type: none"> <li>• Google key configured as default value in parameters</li> <li>• Global context issues for Image Prefix</li> </ul>
	HFix-80828	No error messages while attaching a file to data attachments in PGP UI Kit.

Pega Platform version	Hotfix number	Observed behavior
8.5	HFH-80760	Unexpected error from pzSaveAttachments.
	HFH-69879	Browser hangs when browser refresh is used while user is in Entity home pages in PGP Cosmos application.
	HFH-70536	After user updated the entity details, user details are not refreshed in Entity view.
	HFH-80010	Delegated rules are not getting localized.
8.5	HFH-84884	BAC Pre Registration for activities.
		<p><b>Note:</b> This hotfix has a prerequisite of platform hotfix HFH-84421.</p>

## Pega Government Platform 8.4

The following table lists the required **Pega Government Platform 8.4** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.4.3	HFH-68611	The offline packaging is picking the wrong section

Pega Platform version	Hotfix number	Observed behavior
		rule pyAttachmentScreen . It is picking the rule from the Theme-Clarity application. The Theme-Clarity should be removed from the PGP built-on apps.
	HFix-70417	Unable to resolve the case after opening the case document in inline mode.
8.4	HFix-80895	Google key configured as default value in parameters.
	HFix-67465	SEC update – B20 (PGP 8.4 on Pega Platform 8.4.3)
	HFix-65249	File upload to the external repositories failing with Null Pointer Exception.
	HFix-63547	<ul style="list-style-type: none"> <li>• The Participants parameter configuration is missing for AddPlan flow rule.</li> <li>• Creating a new entity from the Manage association is failing with PegaCRM as built on.</li> <li>• The activity plan option from Mobile</li> </ul>

Pega Platform version	Hotfix number	Observed behavior
		<p>toolkit is invoking the configuration page instead of invoking the plan.</p> <ul style="list-style-type: none"> <li>• PegaGP ruleset was shipped with unlocked ruleset version.</li> </ul>

## Pega Government Platform 8.3

The following table lists the required **Pega Government Platform 8.3** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.3.4	HFix-67402	SEC update – B20 (PGP 8.3 on Pega Platform 8.3.4)
8.3	HFix-57961	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• Visualize in Household view is not working as the ObjectType property was set.</li> <li>• Opening cases from open link of the nodes of the Visualize diagram is not working, for the cases in the implementation layer.</li> </ul>

Pega Platform version	Hotfix number	Observed behavior
		<ul style="list-style-type: none"> <li>The Confirm harness, after the external assignment has been performed is showing Investigation case information - For security reasons we should not show any case information.</li> <li>The data layer attachments are not being saved to CMIS repositories (Fix for SR-D47517 of PGP 8.1).</li> </ul>

## Pega Government Platform 8.2

The following table lists the required **Pega Government Platform 8.2** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.2.8	HFix-68102	SEC update – B20 (PGP 8.2 on Pega Platform 8.2.8)
8.2	HFix-69905	Removing Unique ID generation for person Image if ID doesn't exist while retrieving the person details.

## Pega Government Platform 8.1

The following table lists the required **Pega Government Platform 8.1** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.1.9	HFHix-67795	SEC update – B20 (PGP 8.1 on Pega Platform 8.1.9)
8.1	HFHix-57445	Files not uploading to S3 bucket using PGP framework.
	HFHix-48238	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• Visualize in PGP layer is showing the Subject cases.</li> <li>• While uploading image, the selection of cropped part is not working.</li> <li>• PegaPS ruleset was unlocked.</li> <li>• Upload image functionality for Person or Vehicle is not available in Implementation layers.</li> <li>• Reporting ruleset missing for ICMMangers.</li> <li>• Unable to publishing private message in Pulse section.</li> </ul>

Pega Platform version	Hotfix number	Observed behavior
		<ul style="list-style-type: none"> <li>• Procurement Admin's accessgroup missing the security role.</li> </ul>

## Hotfixes for Pega Government Platform 7.x

- [Pega Government Platform 7.4](#)
- [Pega Government Platform 7.31](#)
- [Pega Government Platform 7.21](#)
- [Pega Government Platform 7.16](#)

## Pega Government Platform 7.4

The following table lists required **Pega Platform** hotfixes for Pega Government Platform 7.4.

Pega Platform version	Hotfix number	Observed behavior
7.4	HFix-46018	Unique IDs are not getting generated on checkbox, if they configured on page list/page group property.
	HFix-45746	The pyLabel property max length issue in pyAttachApprovalEmail activity.

The following table lists required required **Pega Government Platform 7.4** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
7.4	HFix-67796	SEC Update - B20 (PGP 7.4 on Pega Platform 7.4)
	HFix-65487	The entries in the CommList pagelist property were not mapped correctly to the Email and Phone.
	HFix-46678	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• Unable to validate the participants while editing the Interview.</li> <li>• Previous case owner unable to view the investigation details once the ownership is accepted by the new owner.</li> <li>• Unable to delete the multi-line case notes.</li> <li>• Investigation address map is not loading because of the case note address issue.</li> <li>• The end user is able to open the Entity view page from the confirm screen through the link displayed.</li> </ul>

Pega Platform version	Hotfix number	Observed behavior
		<ul style="list-style-type: none"> <li>• Procurements case status is not being updated on rejecting the Review plan.</li> <li>• Application building on ICM are failing because the ICMAadmin accessgorup is referring to PegaDeveloper ruleset in production rulesets.</li> <li>• Issues related to Manage associations fixed.</li> <li>• CustomAudit section is not visible in Subject case on Mobile device.</li> </ul>

## Pega Government Platform 7.31

The following table lists required **Pega Government Platform 7.31** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
8.1	HFix-47979	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• In PGP Selfservice portal, the label and text fields of the username and</li> </ul>

Pega Platform version	Hotfix number	Observed behavior
		<p>passwords are overlapping.</p> <ul style="list-style-type: none"> <li>Branch details are not being displayed in the Business view page.</li> </ul>
7.4	HFHix-67213	The entries in the CommList pagelist property were not mapped correctly to the Email and Phone.
7.3	HFHix-39330	Removing the length restriction in PGP data model.
	HFHix-36427	Broken attachment icons in the submit bid case.
	HFHix-41369	Missing empty hotfix ruleset versions for all the apps.
	HFHix-41488	Hotfix for upgrade PGP 7.31 from UI-Kit 9 to UI-Kit 10.

## Pega Government Platform 7.21

The following table lists required **Pega Government Platform 7.21** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
7.3	HFHix-35637	Delete Document Page error is displayed while logging into the system.
	HFHix-29562	Password mismatch error while creating account.

Pega Platform version	Hotfix number	Observed behavior
7.2.2	HFHix-29562	Password mismatch error while creating account.
7.2.1	HFHix-34240	Saving the PegaPS-Data items is not working as expected.
	HFHix-30345	Mandatory field validations are thrown when user clicks on save.

## Pega Government Platform 7.16

The following table lists required **Pega Foundation for Government 7.16** hotfixes.

Pega Platform version	Hotfix number	Observed behavior
7.3.1	HFHix-35635	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• Opening a case from search complaint encounters an error.</li> <li>• Logging out from the PFG Operator page encounters an error page.</li> <li>• The Create Operator link is not working.</li> <li>• Users cannot view the Add business button in</li> </ul>

Pega Platform version	Hotfix number	Observed behavior
		the Create operator page.
	HFix-30967	Error in the Login screen after clicking on cancel button.
	HFix-29612	Password mismatch error while creating operator.
	HFix-28036	Delete Document Page error is displayed while logging into the system.
7.3	HFix-35635	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• Opening a case from search complaint encounters an error.</li> <li>• Logging out from the PFG Operator page encounters an error page.</li> <li>• The Create Operator link is not working.</li> <li>• Users cannot view the Add business button in the Create operator page.</li> </ul>
	HFix-32809	Support for Pega Certification and Licensing

Pega Platform version	Hotfix number	Observed behavior
7.2.2		for Government 7.21 release.
	HFix-30967	Error in the Login screen after clicking on cancel button.
	HFix-29612	Password mismatch error while creating operator.
	HFix-28036	Delete Document Page error is displayed while logging into the system.
7.2.1	HFix-32809	Support for Pega Certification and Licensing for Government 7.21 release.
	HFix-30967	Error in the Login screen after clicking on cancel button.
	HFix-29612	Password mismatch error while creating operator.
7.2	HFix-30967	Error in the Login screen after clicking on cancel button.
	HFix-28036	Delete Document Page error is displayed while logging into the system.
7.2	HFix-30967	Error in the Login screen after clicking on cancel button.

# Implement

A project team that implements a Pega application can use this implementation guide to enable and extend application features to meet client business requirements. You must complete the product installation before starting the implementation procedures.

For information about installing your application, see the product installation guides on the [Pega Government Platform 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.

- [Preparing for the implementation](#)
- [Testing a new application](#)
- [Packaging an application](#)
- [Production maintenance and monitoring](#)
- [Application data model](#)
- [Pega Government Platform entities](#)
- [Implementing Pega Government Platform features](#)
- [Pega Government Platform case types](#)

- **Time Management**
- **Appendix A: Pega Government Platform Application overview**
- **Appendix B: Creating new entities**
- **Appendix C: Mobile Strategy**
- **Limit the Visibility of sensitive data items through attribute-based access control (ABAC)**
- **Leverage Pega Government Platform from Pega Customer Service**
- **Integrating Industry Applications with Pega Customer Service**
- **Driving UI with Local Data**

## Preparing for the implementation

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

### 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 *PegaGPCosmos:Administrators* 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.
  - d. Save the new Application Setup operator.

2. Log in as the Application Setup operator.
3. In the header of Dev Studio, click the name of the application, and then click New Application.
4. 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](#).

5. On the **Name your application** page, enter the name of the application, and then click Advanced configuration.
6. 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.



**Note:** For the new application, the organization name cannot be PegaPS.

7. Click Save.
8. 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.

9. To open the new application, click Go to app.

**Result:**

The creates a set of access groups for the application. Copy the access roles from the existing access groups and add them to the corresponding implementation access groups. Create your own operators, and then apply the appropriate access groups.

## Implementing the security model

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

### Defining the security model and organization structure

Optionally, define the authorization and authentication strategies for your application.

#### Authentication

Proves to the application that you are who you say you are.

#### Authorization

Determines the functions that you can perform in the application. This corresponds to an access group and role configuration.

Security planning involves defining authorization and authentication strategies for your application. It is a best practice to create new access groups and roles that are based on the default access groups and roles that come with the product.

Security planning also involves setting up the organization structure and operator attributes. The application provides security in the form of access settings and denial rules. Many integration rules also incorporate authentication.

For more information about the additional aspects of security, enroll in the *Lead System Architect* course on Pega Academy.

## 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.

## Defining 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 the 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](#).

## Defining your authorization scheme

Pega Government Platform comes with a predefined set of access groups, roles, and privileges. You can use the application roles as a starting point, but you should create your own application-specific access groups and roles to avoid any future problems when upgrading.

Other rule types such as sections, flow actions, and activities use roles and privileges to allow access to these rules at run time.

### Defining your access groups

Define the access groups that you want to add to your application.

1. Identify additional access groups that are needed for your application.
2. Identify portals associated with these access groups.

For more information, see [Viewing access groups and operators](#).

### Defining access roles and privileges

You can associate one or more roles with an access group. Roles are additive. The more roles that you add to an access group, the more authorization there is. Privileges can be associated with one or more roles.

1. Determine which roles are needed for your application. You can use the Pega Government Platform roles as a starting point.
2. Determine which privileges to associate with each role.
3. Associate each role with an access group.

For more configuration information, see [Groups and roles](#).

## Defining the organization structure

Use the organization structure for routing and reporting within the application. Typically, the application organization structure does not map operators exactly to the site's organization chart but instead, it maps the work that those operators do.



**Tip:** For design guidance, see [Setting up your organization structure](#).

1. In the header of Dev Studio, click **Configure > Org & Security > Organization > Organizational Chart**.
2. Review the existing structure.
3. Determine the organization, division, and unit levels of the hierarchy.

## Defining the operator attributes

An operator's access group affects what the operator can do in the application. In addition to the access group, the following fields in the operator record influence how the application handles assignment of work to the user.



**Tip:** In many implementations, it is more efficient for the application to set values in the operator record during the authentication process than it is to have an administrator manually maintain these records. These rules must be configured as part of the authentication mechanism for your site. For more information, see [Authentication services](#).

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

## Defining the operator work group

The work group setting in the operator record affects how your application delivers work to the operator. Review the Operator record and determine the rules for assigning a work group to an operator or the role that multiple operators hold.

1. In the header of Dev Studio, click **Configure > Org & Security > Organization > Operators.**
2. Select an operator ID.
3. On the **Work** tab, review the work group information for the operator record.
4. Determine your policy for assigning a work group to an operator or the role that multiple operators hold.  
For more information, see [Fields for operator work groups, work queues, and schedules](#).

### Defining the operator skills

Skill settings in the operator record affect how the application routes work to the operator. Skill settings also affect how the application gets the most appropriate work when using the Get Next Work feature. You must determine the skills that are appropriate for your application and operators.

1. Define the skills that are needed for the application.
2. Determine which operator records or roles should be associated with those skills.

### Defining the operator calendar

The application calendar affects date calculations within the application, such as the date between business days calculation, and the SLA goal and deadline date calculation. The calendar on the operator record is relevant only if you have users who are not working in the same time zone as the rest of the organization. Otherwise, the application uses the calendar on the organization record and you can skip this step.

Operator calendars will have an impact on chat availability. If a chat request comes in after or before the business hours defined in the specified calendar, then the requestor receives "Off-hours behavior" message.

1. Determine the calendar instances that are needed for your application.
2. Determine which operator roles need a distinct calendar.
3. Determine the operator location.

For more information, see [Specifying calendar navigation options](#).

## Defining the work groups

A work group determines which work queues you can access.

1. In the header of Dev Studio, click **Configure > Org & Security > Tools > Work Groups**.
2. Review and modify the list if necessary.

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

## Defining the workbaskets

A work queue is a queue of open assignments in the application.

1. In the header of Dev Studio, click **Configure > Org & Security > Tools > Work Queues** to display a list of work queues.
2. Pega Government Platform provides a set of default work queues. Implementation teams must create their own work queues with their respective organization structure. References to the work queues must also be updated accordingly in the implementation layer.

## Defining work parties

A work party represents a person, business, or organization that is involved in a case. It receives correspondence, such as email, and can be an active or passive participant based on its role. Pega Government Platform comes with default work parties, but you might need to configure them for site-specific requirements.

1. In the **Case types** explorer, click the name of the case of which you want to modify the work parties.
2. Click the **Settings** tab and select **Parties**.
3. Click the work party name to open the settings for the work party.
4. Make your modifications and click **OK**.

**What to do next:**

For more information, see [Defining case participants](#).

## Implementing the security model and organization structure

After you review the existing groups and roles to determine additional groups and roles that you need, create them in Dev Studio when logged in as an administrator.

- For Access groups, click **Configure > Application > Structure > Access Groups and Users**.
- For Access roles, click **Configure > Org & Security > Tools > Security > Role Names**.

For more information, see [Learning about access groups](#) and related topics.

## Extending an application built on Pega Government Platform

This section describes the high-level steps for creating an implementation layer on top of an application (such as Investigative Case Management) that is built on Pega Government Platform, then modifying or creating the required rules and structures to support that layer.

After the implementation layer has been created, perform the configurations that are described in the following steps. Ensure that you are logged in as a developer.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the **Switch Studio** menu, and then click **App Studio**.
3. In the navigation pane of App Studio, click **PGP > Application setup**.
4. Click **Setup application**, and then click **Done**.

## Setting up the application

When Application setup is run, it automatically runs the implementation layer steps that are required for the apps that are built on Pega Government Platform, and it performs the following actions.

- Overrides the lookup data pages in base applications work classes to implementation layer specific work classes and updates the source of lookup to the implementation work class.
- Overrides all node level data pages which are in the pattern inheritance of *PegaPS-Data-Context* class to the implementation layer and sets the access group to the implementation layer Admin Access Group. For example, if built-on is *PegaGP*, *PegaPS-Data-Context-Application* is the class. If built-on is ICM then it also considers *PegaPS-Data-Context-ICM* as well as per pattern inheritance.
- Updates the operators of base application with implementation specific access group and sets it to default.
- Updates the implementation Application Data class direct inheritance with the data class of the base application.

Alternatively, teams can perform the steps above manually, if only a subset of the steps is required.

## Testing a new application

Testing a new application involves testing in different environments.

### Testing 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. Use the Performance tool to measure the performance of the application. For more information, see [Track system utilization for a requestor session with Performance Analyzer](#).
  - Prior to extending your site-specific Pega implementation, establish a performance benchmark (baseline) by using the Performance tool. This allows subsequent, iterative performance tests against the baseline to help identify any degradation in performance resulting from development efforts.
  - Use the Performance tool to check the performance of the following features:
    - Search
    - Account selection
    - Loading of components
    - Kickoff of all service intents. Automated scripts are recommended for this unit testing, but are not required.
  - Save the test results so that you can compare them to future test results to determine whether an application update has a performance impact.
3. Verify that the out-of-the-box 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.
4. Test all integrations, both independently and with associated integrations.

Test integrations for any optional Pega Government Platform 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.

5. 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 your 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 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 out-of-the-box 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 Government Platform 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 step 8.
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 it is a difference in environment that causes the test to fail. If the environment causes a failure, either fix the issue that causes 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

preproduction environment. UAT ensures that users will be able to successfully complete their 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 as end-users) run scripts to test all scenarios, including boundary and exception testing. The end-users (trainers, managers, and directors), must then perform the following steps during UAT:
  - a. Verify that there are no major issues.
  - b. Review changes to understand the features.

## Packaging an application

To migrate a new application to a different environment, you must package the application before you can import it to the new environment.

### Merging application changes

If you develop 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.

Additionally, the Pega 1:1 Operations Manager application allows business users to make controlled changes in a Business Operations Environment (BOE), and test the changes in a production environment within boundaries defined by your organization's IT department. For more information, see [Revision management](#).

## Packaging an application for migration

Before you migrate a new application to a different environment, package the relevant data instances and rulesets into a product rule. The product rule is an instance of *Rule-Admin-Product*, which Pega Platform refers to as the RAP file.

### Before you begin:

If you develop your application features in separate branches, merge the branches. For more information, see [Developing applications, merging branches, and deploying changes in a distributed development environment](#).

1. In the header of Dev Studio, click Configure > Application > Distribution > Package to start the **Application Packaging** wizard.  
For more information about using the wizard, see [Packaging your application in a product rule](#).
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. If you want to make any changes, on the last page of the wizard, click Modify.
6. When you have completed your review of the RAP file, click Export.

### Result:

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.

### Before you begin:

Package the relevant data instances and rulesets into a product rule. For more information, see [Packaging an application for migration](#).

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 importing an application to a production environment, see [Deploy Pega Platform](#).

## Production maintenance and monitoring

You need to understand business rule maintenance, application monitoring, and reporting issues.

### Business rule maintenance in the Production environment

You can give managers the ability to update rule types in a production environment. For example, managers can update the Goals and Deadline for a certain case type. You must delegate these rules in Dev Studio first. After you delegate a rule, you can access it in the Interaction Portal by clicking **Profile > My rules**. For more information on rule delegation, see [Delegating a rule or data type](#).

### Monitoring your application

[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 Diagnostic Center 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 your application's 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 times. Over time, slow integration points can cause average handle times to increase. When queues start to grow, it becomes very difficult to recover, and might require the use of offline services or a backup application.

## Reporting issues

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

1. Identify the 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.

## Application data model

The application provides a set of data types, data pages, and sample data, which you can use to begin implementing your application. You will need to use the data from your system of record instead of using the sample data provided by the application.

Data modeling involves creating a conceptual model of how data items relate to each other in an application. In Pega Platform, the data model refers to a set of rules that work together to populate the data in your application. The system displays this data to

help you process the information and can help you make decisions in your sales processes. The following rule types constitute your data model:

### Data types

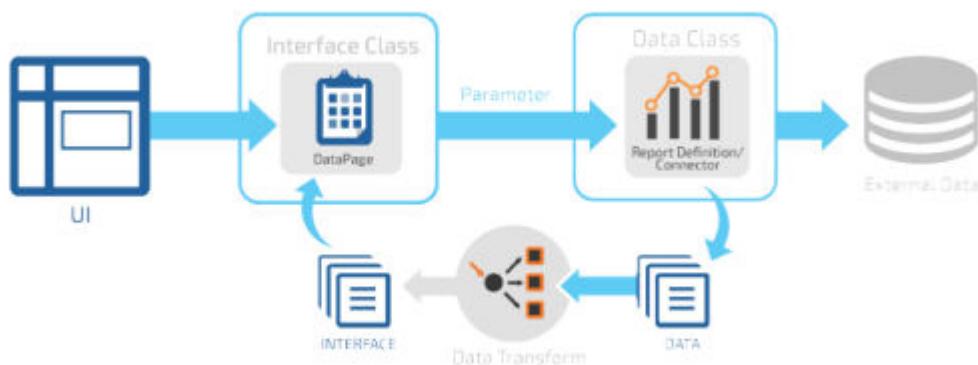
Data type is another name for a class in your application. It holds data that the application uses. A data type has one or more data pages and several property definitions associated with it.

### Properties

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

### Data pages

Data pages define the content of a clipboard page. Data pages also control the loading of that data from a source system, as shown in the following figure.



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.

## Pega Government Platform entities

PGP includes a collection of common government data structures that you can use to collect and persist data in your application. Each entity contains a hierarchy of common scalar properties, embedded page or page list properties, and data instances related by association (foreign-key relationship).

- Person (PegaPS-Data-Entity-Person)

- Business (PegaPS-Data-Entity-Business)
- Vehicle (PegaPS-Data-Entity-Vehicle)
- Facility (PegaPS-Data-Entity-Facility)
- Household (PegaPS-Data-Entity-Household)
- Item (PegaPS-Data-Entity-Item)

Each Pega Government Platform entity data class has supporting case types to provide standardized data management operations on the persistent datastore.

## Add

Collect data and persist to database table.

## Update

Edit data and persist to database table.

## View

Present data that has been read from the database table.

## Search

Collect criteria and retrieve matching instances from database table.

- [Person entity](#)
- [Business entity](#)
- [Vehicle entity](#)
- [Household entity](#)
- [Facility entity](#)
- [Item entity](#)
- [Entity attachments](#)
- [Entity History and Field Audit](#)
- [Entity merge](#)

- **Creating New Entities**

## Person entity

Pega Government Platform provides data support to collect data about people in the form of Person Entity.

The Person Entity feature contains a persistent datastore and three supporting case types to manage the Person Entity datastore.

### Add Person case type

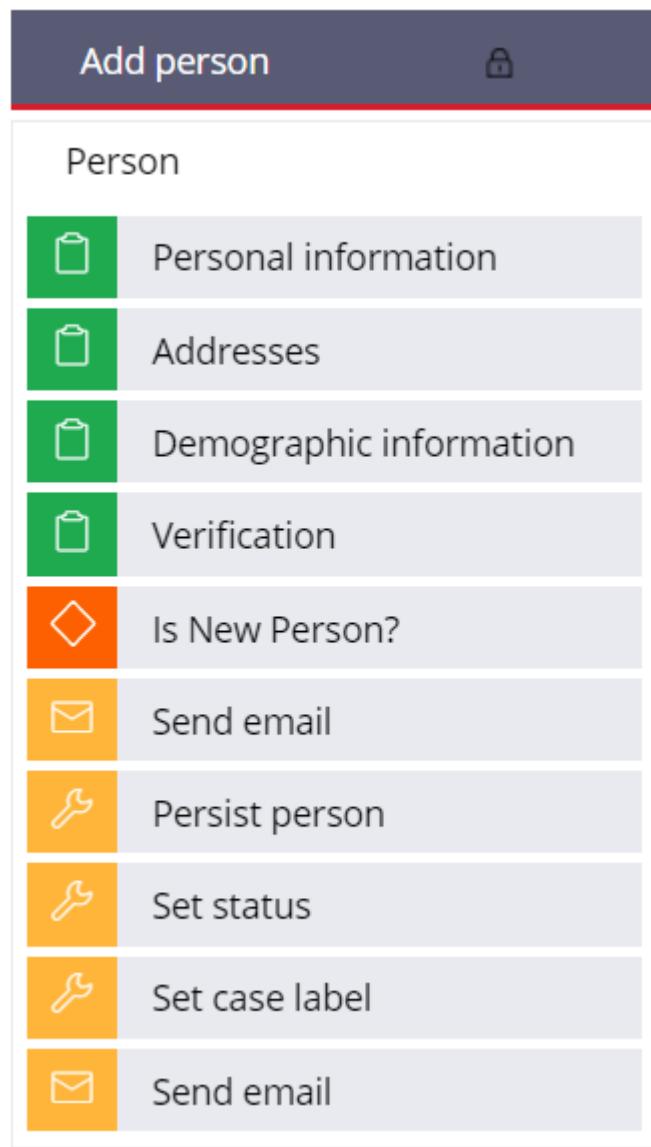
Add a person case type. Using the *PegaPS-Work-Entity-Person-Add* case type, you can introduce a new entity of the type Person into the system.

	Personal information
Addresses	
Demographic information	
Verification	
Is New Person?	
Send email	
Persist person	
Set status	
Set case label	
Send email	

The Add Person case type checks for duplicate person entries and notifies about the same. The case type persists the person record along with the supported data like address, communication, communication preference, profile image, security questions, and so on.

## Adding a new field to the person intake screen

Capture a new field in the person intake screen. You can add a new data field or property and extend the existing person intake screen.



1. In the Person data type, create a field, for example, Spouse Name.
2. Run the Person case type, and then navigate to the PersonBasicInfo view.

3. Click the Add icon, and then add the new field.

The screenshot shows the Pega Platform's configuration interface. On the left, a 'New Person' form is displayed with various fields like First name, Middle name, Last name, Date of birth, Marital status, etc. A new field, 'Spouse Name', is highlighted with an orange border. On the right, the 'Template' view shows the structure of the form, including a section labeled 'A' containing 'First Name', 'Middle name', 'Last Name', 'Date of birth', 'Marital status', 'Other classification', 'ID Number', 'Tax ID', 'Resident status', and 'Permanent resident ID'. A 'Change' button is visible above the template list.

### Result:

The new field is automatically saved in the person table.

## Updating duplicate check validation for a field

Update the existing duplicate check fields with new fields.

1. Open the *CheckDuplicate (PegaPS-Data-Entity-Person)* case match rule.
2. Add a new field by clicking Add must match condition.

3. Update the filter logic accordingly.

**Case Match: Check Duplicate [ Available ]**  
**CL:** PegaPS-Data-Entity-Person ▾ **ID:** CheckDuplicate **RS:** PegaPS:08-05-01

Evaluation Pages & Classes Specifications History

Case Matching Type  
Seek cases

**Must match conditions**

Conditions that must evaluate to true for case to be considered a duplicate

Label	Potential duplicates	Relationship	Current case
C	.Name.FirstName	is same	
A	.Name.LastName	is same	
B	.DOB	is same	
D	.MergeStatus	is equal	"Active"
E	.MergeStatus	is null	

+ Add must match condition

Hide filter logic C AND A AND B AND (D OR E)

4. Click Save as.  
 5. In the Context section, select an application.  
 6. In the Add to ruleset area, select a ruleset and the ruleset version.  
 7. Click Create and open.

## Adding a new list item

Update an existing list of items with new items.

Add a new list item, for example, GrantsList and persist it.

1. In Dev Studio, in the PegaDATA database, create the Grant data type with the following properties: *pyID* (Primary key), *EntityID*, and *EntityType*.
2. Add other properties that are related to the Grant, such as the grant name, expiry date, and so on.
3. In the Person data type (*PegaPS-Data-Entity-Person*), create a field of type page list of Grant data type class.
4. Update the *PersistEmbeddedPage* (*PegaPS-Data-Entity-Person*) activity with a new step that loops the *GrantsList* property, and then invoke the activity *SaveObject* by passing the *EntityID* and *EntityType* parameters similar to *CommList*.

Label	Method	Step page	Description
1.	Loop When > Call LoadAddressListLatLong		Invoke activity to load GeoCodeAPI for address
2.	Loop When > Call SaveObject	.CommList	Save Communications

**Method Parameters**

Pass current parameter page

Name	Value
*EntityID	Primary.pyID
*EntityType	Primary.EntityType

5. Create a UI to capture the Grant details.

## Update Person case type

Update the existing *Person* case type. Using the *PegaPS-Work-Entity-Person-Update* case type, you can update the details of an existing entity.

The screenshot shows the 'Update person' process configuration page. At the top, there's a header bar with the title 'Update person'. Below it, a section titled 'Person' contains several items:

- Personal information (green icon)
- Addresses (green icon)
- Demographic information (green icon)
- Verification (green icon)
- Is New Person? (orange diamond icon)
- Send email (orange envelope icon)
- Persist person (orange key icon)
- Set status (orange wrench icon)
- Set case label (orange wrench icon)
- Send email (orange envelope icon)

At the bottom of the list is a button labeled 'CONFIGURE PROCESS'.

You can make changes to the person data and submit them. The details persist after the duplicate validation check ensures that there are no duplicates.

The *D\_Person* data page fetches the entity details along with associated data like address, communication, and so on.

## Updating an existing list property

Update an existing list property, for example, GrantsList.

You can update the existing list properties for the person entity. Fetch the grant details to update the existing list of grants that are captured as a part of the Add Person procedure.

1. Update the *FetchPersonDetails (PegaPS-Data-Entity-Person)* data transform.
2. Add a step similar to CommList or AddressList to set the details using the corresponding grant data page of type: List.

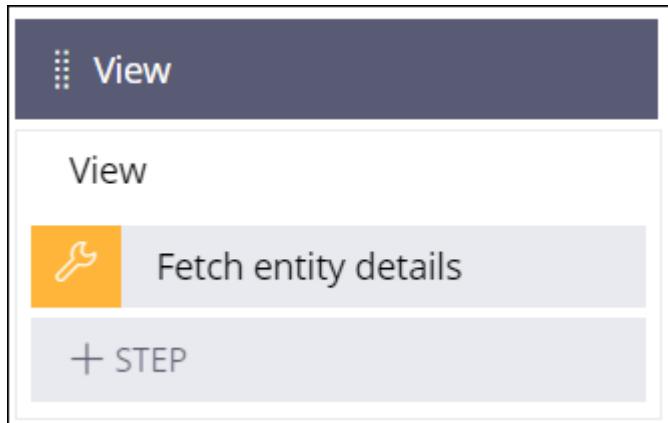
Step	Action	Parameter	Condition	Target
• 14	Set	Param.a	equal to	@Utilities.pxExecuteAnA
• 15	When	Param.BasicDetails		
• 16	Otherwise			
• 16.1	Set	.CommList	equal to	D_CommunicationList[ID]
• 16.2	Set	.AddressList	equal to	D_AddressList[ID:Param]
• 16.3	Set	.SecurityQuestionList	equal to	D_QuestionList[ID:Param]
• 16.4	Set	.IdentifiersList	equal to	D_IdentifiersList[ID:Param]

### Result:

The Update case populates the details and any changes made are persisted.

## View Person case type

View the details of a person entity. Using the *PegaPS-Work-Entity-Person-View* temporary case type, you can view the complete details of an existing person entity.



The *EntityDetails (PegaPS-Data-Entity-Person)* wrapper section contains all the person related views or sections.

The screenshot displays the Pega Studio interface. On the left is a person view for 'Johnny Flores' showing various personal details like ID, phone numbers, email, address, and contact preferences. In the center, a 'Section' panel is open under the 'Details' tab, displaying a grid of personal information such as Full name, Date of birth, Marital status, Gender, Eye color, Ethnicity, Tax ID, Resident status, Permanent resident ID, Other classification, and Active military status. To the right, the 'Entity details' template is being configured. It shows a visual representation of the layout with a single column and a section labeled 'A'. The configuration pane lists components like 'Button Template', 'Person basic details read ...', and 'Security question', each with their respective settings.

## Adding a field to the person view

Add new fields to an existing person view.

Add a field, for example, Spouse Name to the person view.

- To add the field, navigate to the *EntityDetails* wrapper section and choose the appropriate section to show the field, for example, the Entity details section.

The screenshot shows the Pega Studio interface. On the left is a preview of a person record for 'Johnny Flores'. The 'Details' section is expanded, showing fields like ID, Home phone, Mobile phone, Email, Home address, Mailing contact preference, Contact time, Language, Created by, and Created date. To the right of the preview is a configuration pane for the 'Entity details' section. The 'Template' tab is selected, showing a 1-column layout. The 'Settings' tab is also visible. Below the template area, there are sections for 'Button Template', 'Person basic details read ...', and 'Security question'.

- Add the new field along with the existing ones.

## Showing captured list details in the person view

Show the captured list, for example, GrantsList details in the person view.

- Create a view or section in the *PegaPS-Work-Entity-Person-View* class.
- Include the section in the Main Content region of the *pyCaseMainInner* view.

The screenshot shows the Pega Studio interface. On the left is a preview of a person record for 'Johnny Flores'. The 'Details' section is expanded, showing fields like ID, Home phone, Mobile phone, Email, Home address, Mailing contact preference, Contact time, Language, Created by, and Created date. To the right of the preview is a configuration pane for the 'pyCaseMainInner' view. The 'Main content' region is selected and contains a 'Utils' section. The right side shows a list of available sections: 'Fixed summary pane', 'Case Header', 'Case Action Header', 'Scrollable summary pane', 'Case details', 'Main content (Grouped - Defau...', 'Case Information', 'Experience view wrapp...', 'Education view wrapper', 'Training view wrapper', 'My Cases List', 'Activity page', 'Utils', and 'Association utility'.

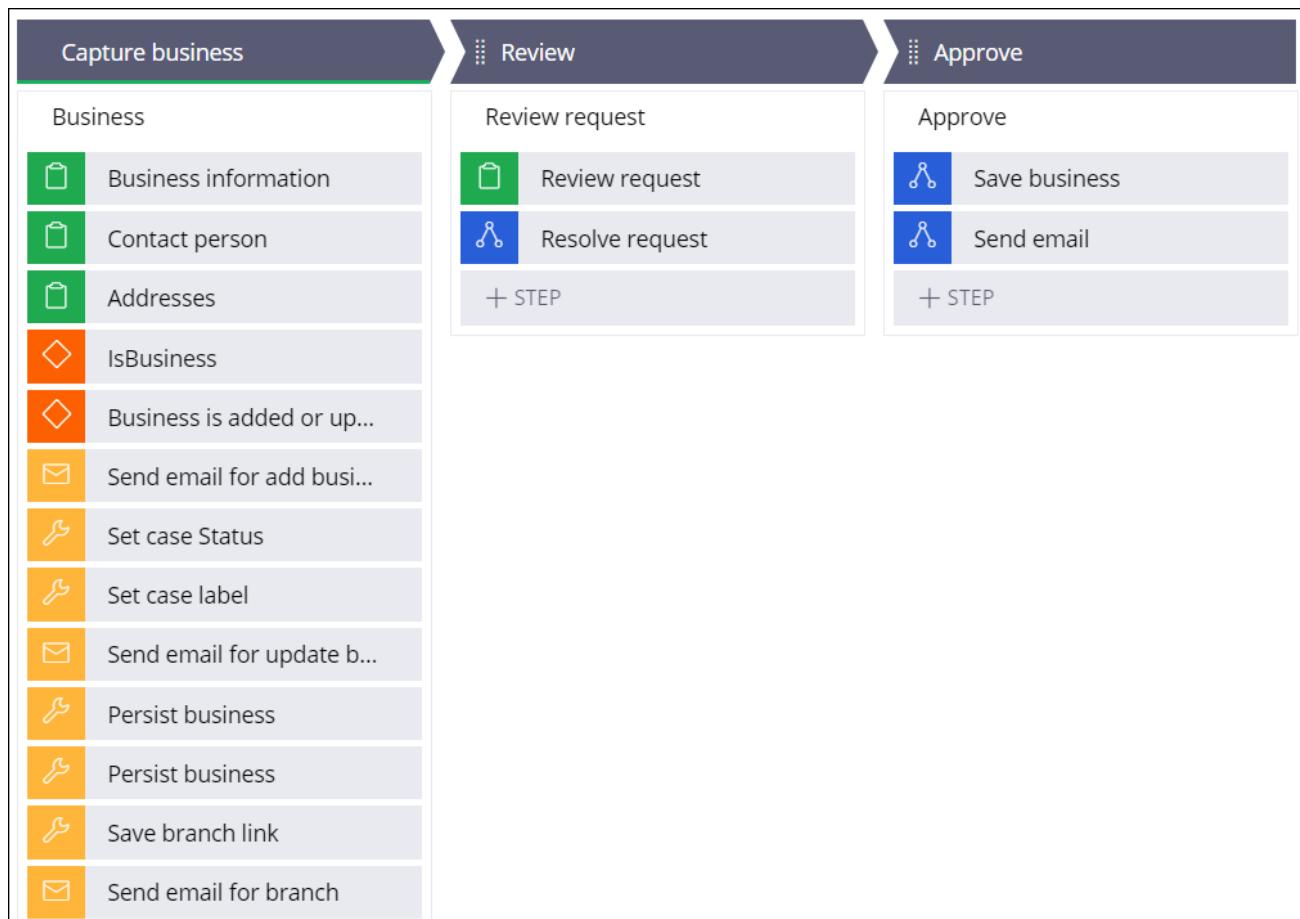
## Business entity

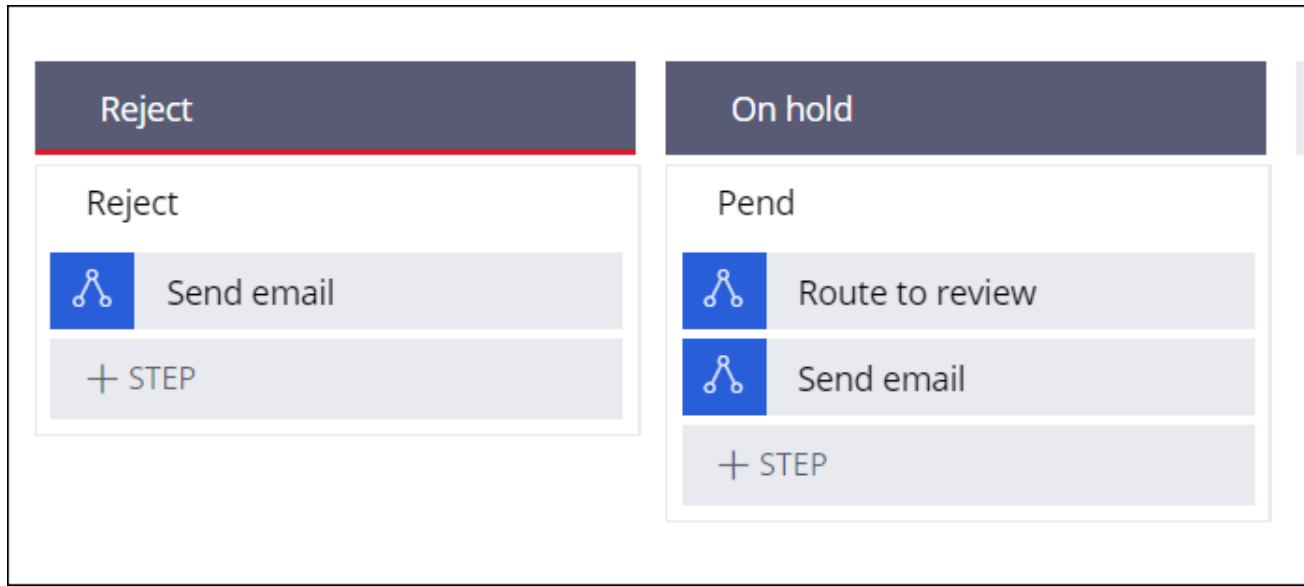
Pega Government Platform provides data support to collect data about business or organizations in the form of Business Entity.

The Business Entity feature contains a persistent datastore and three supporting case types to manage the Business Entity datastore.

### Add Business case type

Add a business case type. Using the *PegaPS-Work-Entity-Business-Add* case type, you can introduce a new entity of the type Business into the system.





The Add Business case type checks for duplicate business entries and notifies about the same. The manager has the options: Approve, On hold, and Reject.

**B** Review request  
Due 2 days from now

Take action \*

Approve       On hold       Reject

**Cancel** **Save** **Submit**

After the manager approves, the business details are persisted along with supported data like address, communication, profile image, and so on.

## Adding a new field to the business intake screen

Capture a new field in the business intake screen and persist it. You can add a new data field or property and extend the existing business intake screen.

1. Create a field, for example, Parent Organization, in the business data type.
2. Run the Add Business case type and navigate to the BusinessBasicInfo view.

- Click the Add icon and add the field created in step 1.

The screenshot shows the Pega Government Platform Cosmos interface. On the left, the 'New Business' form is displayed with several fields grouped under a 'Section' header, which is highlighted with an orange border. These fields include 'Trade name \*', 'Legal name \*', 'Employer ID \*', 'Ownership type \*', 'Open date', 'Number of employees', 'Business email \*', 'Business phone', 'Alternate phone', and 'Business fax'. Below the form is a 'Continue' button. To the right of the form is the 'BusinessBasicInfo' template editor. The template editor shows a grid layout labeled '3 Column (Inline Wrapping)'. It lists various fields with their data types: Trade name (Text input), Legal name (Text input), Employer ID (Text input), Ownership type (Dropdown), Open date (Dropdown), Year started (Dropdown), Branch name (Text input), Branch ID (Text input), Open date (Dropdown), and Year started (Dropdown). The 'Template' tab is selected.

### Result:

The new field automatically gets persisted in the business table.

## Updating duplicate check validation for a field

Update the existing duplicate check fields with new fields.

- Open the *CheckDuplicate (PegaPS-Data-Entity-Business)* case match rule.
- Add a new field by clicking Add must match condition.

3. Update the filter logic accordingly.

Label	Potential duplicates	Relationship	Current case
A	.EIN	is same	

+ Add must match condition

Hide filter logic A

4. Click Save as.
5. In the Context section, select an application.
6. In the Add to ruleset area, select a ruleset and the ruleset version.
7. Click Create and open.

## Adding a new list item

Update an existing list of items with new items.

Add a new list item, for example, SubsidiaryList and persist it.

1. In Dev Studio, to add a new list item, create a data type Subsidiary in the PegaDATA database and have the following properties pyID (Primary key), EntityID, and EntityType.
2. Add other properties related to subsidiary like subsidiary name, incorporated date, and so on.
3. In the Business data type (*PegaPS-Data-Entity-Business*), create a field.
  - a. In the Field name box, enter SubsidiaryList.
  - b. In the Type list, select Embedded data.
  - c. In the Data object list, select Subsidiary.

d. Click Submit.

Add field to Business

Field name \*

SubsidiaryList

Type

Embedded data

Data object \*

Subsidiary

Options

Single record  
 List of records

> Advanced

Cancel      Submit & add another      Submit

4. Update the *PersistEmbeddedPage* (*PegaPS-Data-Entity-Business*) activity with a new step looping the SubsidiaryList property and invoke the activity *SaveObject* by passing the EntityID and EntityType parameters similar to CommList.

Activity: Persist embedded page [Available]

CL: PegaPS-Data-Entity-Business ID: PersistEmbeddedPage RS: PegaPS:08-06-01

This record has 1 info warning (including 1 unjustified) View

Steps Parameters Pages & Classes Security Test cases Specifications History

Label	Method	Step page	Description
1.	Loop When > Call LoadAddressListLatLong		Invoke activity to load GeoCodeAPI for address: Jump
2.	Loop When > Call SaveObject	.CommList	Save Communications Jump

Method Parameters

Pass current parameter page

Name	Value
*EntityID	Primary.pyID
*EntityType	Primary.EntityType

5. Create a UI to capture the subsidiary details so that they persist after you enter the data.

## Update Business case type

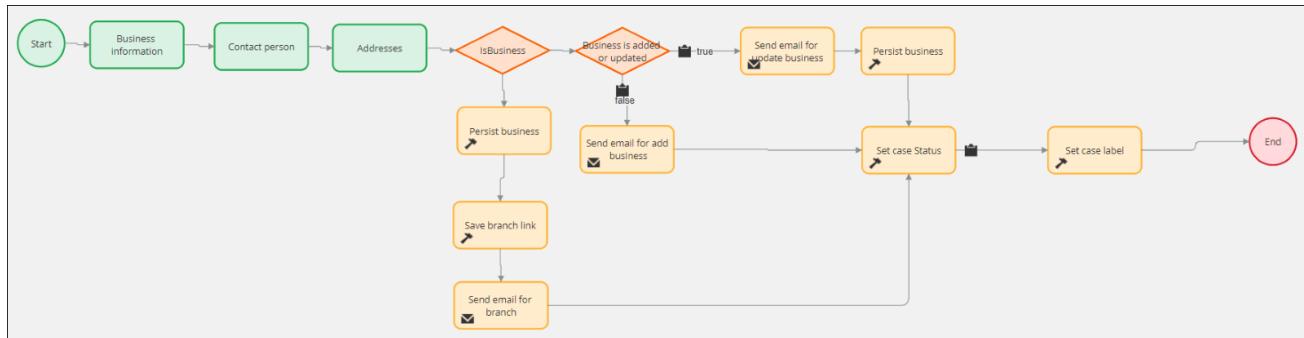
Update the existing business case type. Using the *PegaPS-Work-Entity-Business-Update* case type, you can update the details of an existing entity.

**Update**

Business

-  Business information
-  Contact person
-  Addresses
-  IsBusiness
-  Business is added or up...
-  Send email for add busi...
-  Set case Status
-  Set case label
-  Send email for update b...
-  Persist business
-  Persist business
-  Save branch link
-  Send email for branch

The Business flow supports updating the Business and Branch details as well.



You can make changes to the Business data and submit them. The details persist after the duplicate validation check ensures that there are no duplicates.

The *D\_Business* data page fetches the entity details along with associated data like address, communication, and so on.

## Updating an existing list property

Update the existing list properties for the business entity.

Fetch the subsidiary details to update the existing list of subsidiaries that are captured as a part of Add Business. Update an existing list property, for example, SubsidiaryList.

1. Update the *FetchBusinessDetails* (*PegaPS-Data-Entity-Business*) data transform.
2. Add a step similar to CommList or AddressList to set the details using the corresponding subsidiary data page of type: List.

The screenshot shows the Pega Studio Data Transform editor for the 'FetchBusinessDetails' transform. The interface includes a toolbar with tabs like Home, D\_Business, LoadBusinessD..., and FetchBusiness...; a top bar with Save as, Actions, and Private edit; and a main table area with rows of steps. The table has columns for Step ID, Type, Field, Operator, Value, and Options.

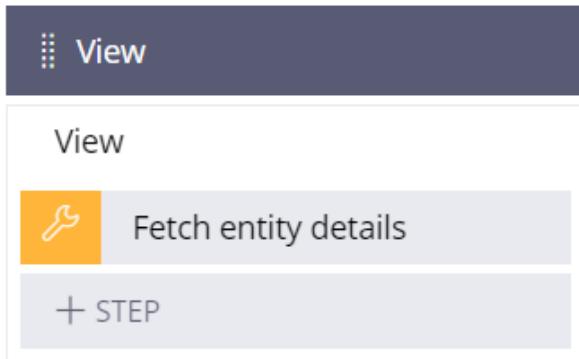
Step ID	Type	Field	Operator	Value	Options
3	Otherwise				
3.1	Set	.BranchList	equal to	D_BranchList[ID:EntityId]	Select values +
3.2	Remove	Primary.BranchAddressList			
3.3	For Each Page I	.BranchList			<input checked="" type="checkbox"/> Also use each page as source context
3.4	Apply Data Tran	InitializeContactPerson			
3.5	For Each Page I	.ContactPerson.CommList			<input type="checkbox"/> Also use each page as source context
3.6	Set	.AddressList	equal to	D_AddressList[ID:Param]	Select values +
3.7	Set	.CommList	equal to	D_CommunicationList[]	Select values +

**Result:**

The Update case populates the details and any changes made are persisted.

## View Business case type

View the details of a business entity. Using the *PegaPS-Work-Entity-Business-View* temporary case type, you can view the complete details of an existing business entity.



The Details (*PegaPS-Data-Entity-Business*) wrapper section contains all the business-related views or sections.

The screenshot displays a business view for 'Morgan Consolidated' on the left, showing various details like ID, phone numbers, and address. The main content area shows 'Details' and 'Locations' sections. The 'Locations' section includes a map of Dothan, Alabama, with a red pin indicating the office address. On the right, there's a configuration interface for a template named 'A', which contains a single column with 100% width content. The template structure is shown with sections for 'Entity Details', 'Location', and 'Business Location'.

## Adding a field to the business view

Add new fields to an existing business view.

Add a field, for example, Parent Organization to the business view.

1. To add the field, navigate to the Basic details RO wrapper section and choose the appropriate section to show the field, for example, the Main details RO section.

Section	A	B	
Trade name	Morgan Consolidated	Open date	Feb 1941
Legal name	Morgan Consolidated	No of employees	987
Employer ID	E132	Non-profit	Yes
Ownership type	Limited liability corporation	Inside city limits	No

2. Add the new field along with the existing ones.

## Showing captured list details in the business view

Show the captured list, for example, SubsidiaryList details in the business view.

1. Create a view or section in the *PegaPS-Work-Entity-Business-View* class.
2. Include the section in the main content region of *pyCaseMainInner* view.

The screenshot displays the Pega Government Platform interface. On the left, there is a 'Workarea' sidebar with a 'Fixed summary pane' header. It contains a logo for 'Morgan Consolidated' and sections for 'Edit' and 'Actions'. Below these are several data fields: ID (BUSINESS-104), Business phone ((123) 456-7890), Alternate phone ((234) 567-8906), Business fax (0987654321), Business email (Morgan.Consolidated@domain.com), Office address (557 N Crescent Street Dothan Illinois 78786 United States), Mailing address (Same as above), Contact person (Sai Shankar), Role/TITLE (---), Email (Morgan.Consolidated@domain.com), and Emergency phone (---). A 'Details' button is at the bottom.

The main content area shows a 'Main content' section titled 'Details' with a purple circular icon containing 'MC'. It lists company information: Trade name (Morgan Consolidated), Legal name (Morgan Consolidated), Employer ID (E132), Open date (Feb 1941), No. of employees (987), Non-profit (Yes), Ownership type (Limited liability corporation), Inside city limits (No).

Below the details is a 'Locations' section with a map of Dothan, Alabama. The map shows the location of 'Niregrass Museum of Art' and 'Dothan'. Two pins are placed on the map, each with a location card: one for '557 N Crescent Street Dothan Illinois 78786 United States' (Office address, Mapped) and another for '557 N Crescent Street Dothan Illinois 78786 United States' (Mailing address, Mapped).

The right side of the interface features a vertical sidebar titled 'Py case main inner'. It includes sections for 'Case Header' (Section), 'Case Action Header' (Section), 'Case details' (Section), 'Case Information' (Section), 'My Cases List' (Section), 'Activity page' (Section), and 'Utils' (Section). Under 'Utils', there are three items: 'Branchlist utility' (Section), 'Association utility' (Section), and 'Document' (Section). The 'Utils' section has a count of 1 item.

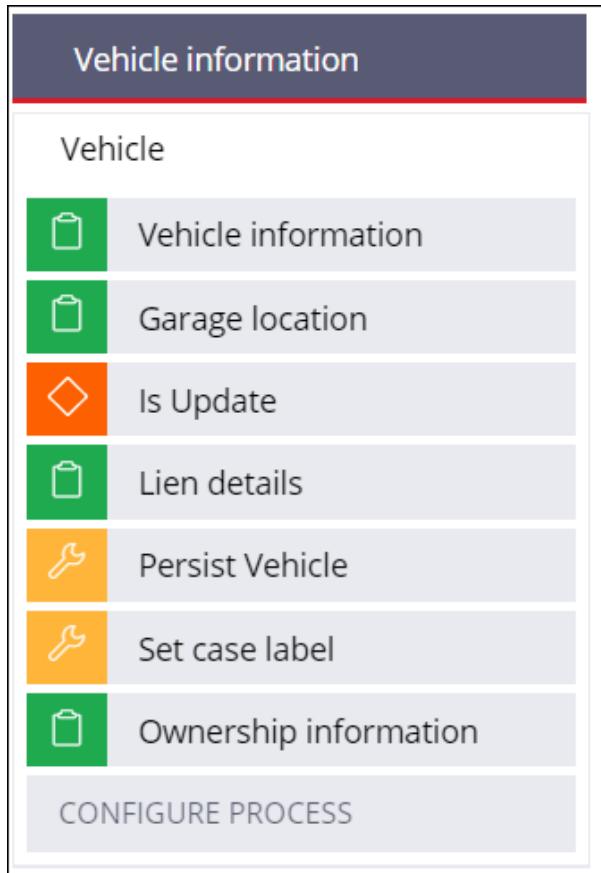
## Vehicle entity

Pega Government Platform provides data support to collect data about vehicles in the form of vehicle entity.

The Vehicle Entity feature contains a persistent datastore and three supporting case types to manage the Vehicle Entity datastore.

### Add Vehicle case type

Add a vehicle case type. Using the *PegaPS-Work-Entity-Vehicle-Add* case type, you can introduce a new entity of the type Vehicle into the system.



The Add Vehicle case type checks for duplicate vehicle entries based on the Vehicle Identification Number (VIN) and notifies about the same. The case type persists the vehicle details along with the supported data like address, owner, profile image, lien details, and so on.

## Adding a new field to the vehicle intake screen

Capture a new field in the vehicle intake screen and persist it. You can add a new data field or property and extend the existing vehicle intake screen.

1. Create a field, for example, Purchase Date, in the vehicle data type.
2. Run the Vehicle case type and navigate to the Vehicle details view.
3. Click the Add icon and add the field created in step 1.

The screenshot shows the Pega Government Platform interface. On the left, the main dashboard has a 'Post' button and a 'Start a con...' button. The central part displays a 'New Vehicle' form with tabs for 'Vehicle information', 'Garage location', 'Ownership information', and 'Lien details'. The 'Vehicle information' tab is active, showing fields for Make\*, Model\*, Manufacture year\*, Vehicle identification number\*, Vehicle registration number, Vehicle registration issuer, Color, Occupant capacity, and Vehicle type (dropdown). A 'Continue' button is at the bottom right. To the right of the form is a 'Vehicle details' panel with a 'Template' section showing a 3 Column (Inline Wrapping) layout and a list of fields with their types: Make (Text input), Model (Text input), Manufacture year (Number), Vehicle identification n... (Text input), Vehicle registration nu... (Text input), Vehicle registration iss... (Text input), Color (Text input), Occupant capacity (Integer), and Vehicle type (Dropdown).

### Result:

The new field automatically gets persisted in the vehicle table.

## Updating duplicate check validation for a field

Update the existing duplicate check fields with new fields.

1. Open the *CheckDuplicate (PegaPS-Data-Entity-Vehicle)* case match rule.
2. Add a new field by clicking Add must match condition.

3. Update the filter logic accordingly.

Label	Potential duplicates	Relationship	Current case
A	.VIN	is same	

4. Click Save as.
5. In the Context section, select an application.
6. In the Add to ruleset area, select a ruleset and the ruleset version.
7. Click Create and open.

## Adding a new list item

Update an existing list of items with new items.

Add a new list item, for example, ServiceHistoryList and persist it.

1. In Dev Studio, to add a new list item, create a data type ServiceHistory in the PegaDATA database and have the following properties pyID (Primary key), EntityID, and EntityType.

2. Add other properties related to subsidiary like service center name, service date, and so on.
3. In the Vehicle data type (*PegaPS-Data-Entity-Vehicle*), create a field.
  - a. In the Field name box, enter ServiceHistoryList.
  - b. In the Type list, select Embedded data.
  - c. In the Data object list, select ServiceHistory.
  - d. Click Submit.

**Add field to Vehicle**

Field name *	ServiceHistoryList
Type	Embedded data
Data object *	Select...
Options	<input type="radio"/> Single record <input checked="" type="radio"/> List of records
<a href="#">Advanced</a>	

4. Update the *PersistEmbeddedPage* (*PegaPS-Data-Entity-Vehicle*) activity with a new step looping the ServiceHistoryList property and invoke the activity SaveObject by

passing the EntityID and EntityType parameters similar to CommList.

Label	Method	Step page	Description
1. [ ] Loop When > [ ]	Call LoadAddressListLatLong	[ ]	Invoke activity to load GeoCodeAPI for address: [ ]
2. [ ] Loop When > [ ]	Call RemoveObject	[ ]	Remove the deleted lien details from the data: [ ]
3. [ ] Loop When > [ ]	Call SaveObject	[ ].LienList	Save Lien details [ ]

**Method Parameters**

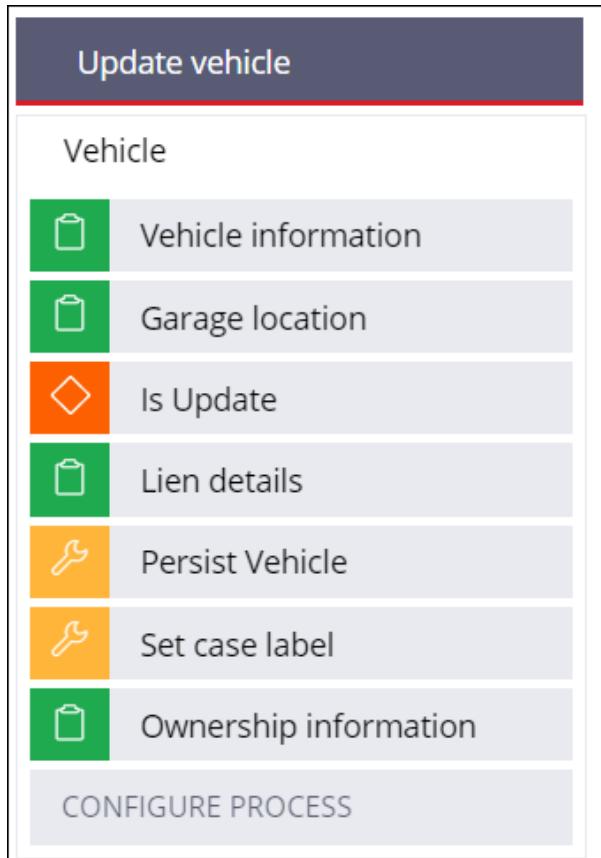
Pass current parameter page

Name	Value
* EntityID	Primary.pyID
* EntityType	Primary.EntityType

5. Create a UI to capture the service history details so that they persist after you enter the data.

## Update Vehicle case type

Update the existing vehicle case type. Using the *PegaPS-Work-Entity-Vehicle-Update* case type, you can update the details of an existing entity.



The Update vehicle case type fetches the details of the Vehicle entity that needs to be updated. You can make changes to the vehicle data and submit them. The details persist after the duplicate validation check ensures that there are no duplicates.

The *D\_Vehicle* data page fetches the entity details along with associated data like address, lien, and so on.

## Updating an existing list property

Update the existing list properties for the household entity.

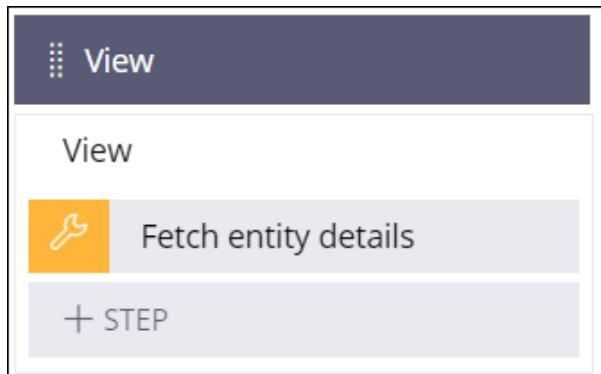
Fetch the subsidiary details to update the existing list of subsidiaries that are captured as a part of Add Vehicle. Update an existing list property, for example, ServiceHistoryList.

1. Update the *FetchVehicleDetails (PegaPS-Data-Entity-Vehicle)* data transform.
2. Add a step similar to AddressList.

	Action	Target	Relation	Source	
• 1	Comment	Used as a data source for D_Vehicle data page			<span style="color: blue;">Delete</span>
• 2	Set	.AttachmentRefID	equal to	.pxObjClass + " " + .Enti	<span style="color: blue;">Select values +</span> <span style="color: blue;">Edit</span> <span style="color: blue;">Delete</span>
• 3	Set	.AddressList	equal to	D_AddressList[ID:Param]	<span style="color: blue;">Select values +</span> <span style="color: blue;">Edit</span> <span style="color: blue;">Delete</span>

## View Vehicle case type

View the details of a vehicle entity. Using the *PegaPS-Work-Entity-Vehicle-View* temporary case type, view the complete details of an existing vehicle entity.



The Details (*PegaPS-Data-Entity-Vehicle*) wrapper section contains all the vehicle related views or sections.

The screenshot shows the Pega Government Platform interface. On the left, a 'Workarea' sidebar displays a vehicle record for a '2016 Mercedes GLA'. The main content area shows a 'Details' section with vehicle information like Make (Mercedes), Model (GLA), Year (2016), and Type of vehicle (SUV). Below this is a 'Locations' section showing a map of Chantilly, Virginia, with a pin indicating the vehicle's location at 4134 Stringfellow Road, Fairfax, Virginia 22033. On the right, a 'Template' editor window is open, showing a visual representation of the page structure with sections labeled 'A' and 'B'. The template details show '1 Column with 100% width content'.

## Adding a field to the vehicle view

Add new fields to an existing vehicle view.

Add a field, for example, Purchase Date to the vehicle view.

1. To add the field, navigate to the Details wrapper section and choose the appropriate section to show the field, for example, the Vehicle header RO section.

The screenshot shows a vehicle detail view for a '2016 Mercedes GLA'. The main pane displays vehicle details such as Make (Mercedes), Model (GLA), Year (2016), and Location (4134 Stringfellow Road Fairfax, Virginia 22033 United States Garage). A sidebar on the right shows the template structure for the 'VehicleInfoRO' section, which includes two columns for inline wrapping.

2. Add the new field along with the existing ones.

## Showing captured list details in the vehicle view

Show the captured list, for example, ServiceHistoryList details in the vehicle view.

1. Create a view or section in *PegaPS-Work-Entity-Vehicle-View* class.
2. Include the section in the Main content region of the *pyCaseMainInner* view.

The screenshot shows a vehicle detail view for a '2016 Mercedes GLA'. The main pane displays vehicle details such as Make (Mercedes), Model (GLA), Year (2016), and Location (4134 Stringfellow Road Fairfax, Virginia 22033 United States Garage). A sidebar on the right shows the template structure for the 'Case main inner sect...' section, which includes a fixed summary pane, case header, and scrollable summary pane.

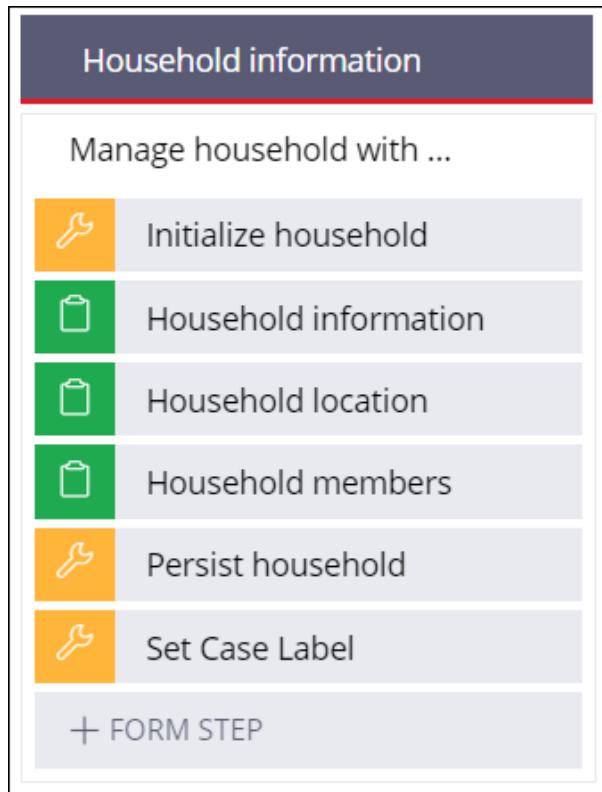
## Household entity

Pega Government Platform provides data support to collect data about households in the form of Household Entity.

The Household Entity feature contains a persistent datastore and three supporting case types to manage the Household Entity datastore.

### Add Household case type

Add a household case type to the system. Using the *PegaPS-Work-Entity-Household-Add* case type, you can introduce a new entity of the type Household into the system.



Adding a new household is a three step process. In the Household information step, the basic details like name, type of household, name, description, and start date are

captured. In the next step, address details are captured and in the last step details of the household members are captured, and then all the details are persisted.



**Note:** Unlike other Pega Government Platform entities, household does not have any duplicate validation check.

## Adding a new field to the household intake screen

Capture a new field in the household intake screen and persist it. You can add a new data field or property and extend the existing household intake screen.

1. Create a field, for example, Is Remote location in the household data type.
2. Run the Household case type and navigate to the Capture household information view.
3. Click the Add icon and add the field created in step 1.

The screenshot shows the Pega Government Platform Cosmos interface. On the left, the 'New Household' form is displayed with sections for 'Household information', 'Household location', and 'Household members'. The 'Household information' section is highlighted with an orange border. It contains fields for 'Household name \*' (text input), 'Household type \*' (dropdown), 'Dwelling type' (dropdown), 'Start date \*' (date time input), and 'Description' (text area). At the bottom of this section are 'Cancel' and 'Continue' buttons. On the right, the 'Capture household in...' template editor is shown. It includes a 'Template' section with a 3 Column (Inline Wrapping) layout, a 'Settings' section, and a list of fields under section 'A': Household name (Text input), Household type (Dropdown), Dwelling type (Dropdown), Start date (Date time), and Description (Text area).

### Result:

The new field automatically gets persisted in the household table.

## Adding a new list item

Update an existing list of items with new items.

Add a new list item, for example, HouseholdItemsList and persist it.

1. In Dev Studio, to add a new list item, create a data type HouseholdItems in the PegaDATA database and have the following properties pyID (Primary key), EntityID, and EntityType.
2. Add other properties related to subsidiary like item name, cost, purchased on, and so on.
3. In the Vehicle data type (*PegaPS-Data-Entity-Household*), create a field.
  - a. In the Field name box, enter HouseholdItemsList.
  - b. In the Type list, select Embedded data.
  - c. In the Data object list, select HouseholdItem.
  - d. Click Submit.

The screenshot shows the 'Add field to Household' dialog box. It has the following fields:

- Field name \***: HouseholdItemsList
- Type**: Embedded data
- Data object \***: Select...
- Options**:
  - Single record
  - List of records
- Advanced**: A link to more options.

4. Update the *PersistEmbeddedPage* (*PegaPS-Data-Entity-Household*) activity with a new step looping the HouseholdItemsList property and invoke the activity SaveObject

by passing the EntityID and EntityType parameters similar to AddressList.

The screenshot shows the Pega Platform's activity editor. The activity is named "Persist embedded page [Available]" and is set to "Private edit". The activity details pane shows a sequence of steps:

- Loop When > Call LoadAddressListLatLong
- Loop When > Property-Set
- Loop When > .MembersList
- Loop When > Connect-Wait
- Loop When > Call SaveObject

Method Parameters:

- Pass current parameter page

Name	Value
* EntityID	Primary.pyID
* EntityType	Primary.EntityType

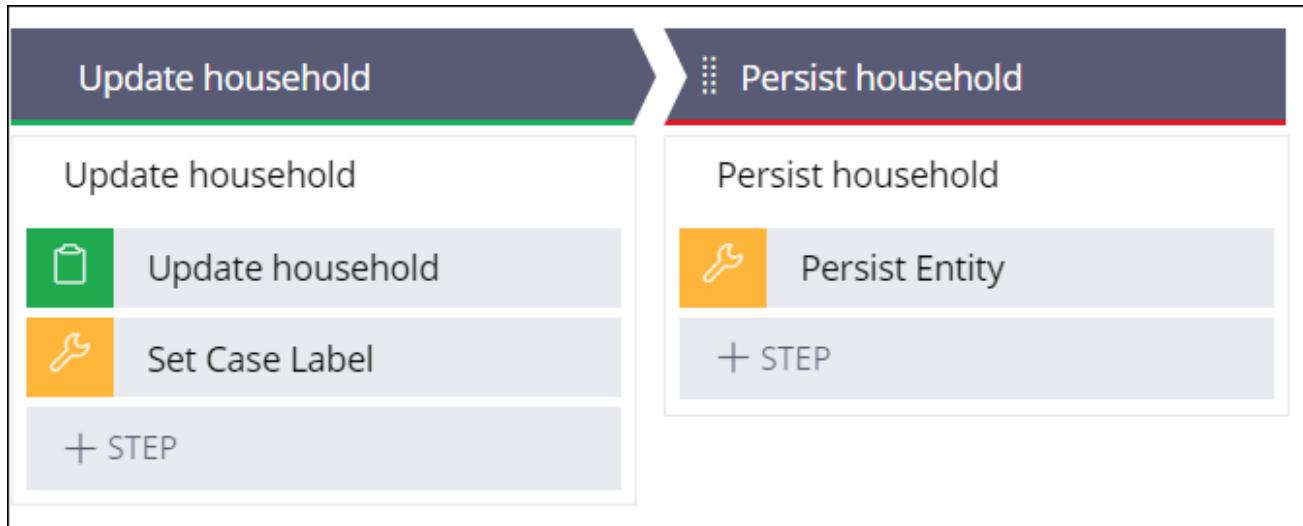
5. Create a UI to capture the household item details so that they persist after you enter the data.

## Update Household case type

Update the existing household case type. Using the *PegaPS-Work-Entity-Household-Update* case type, you can update the details of an existing entity.

**Note:** In this case type, you can update only the basic household details, but

- ⓘ not the members of the household. The Manage Members case type supports updating household members.



The Update household case type fetches the details of the Household entity that needs to be updated. You can make changes to the household data and submit them. The details persist after you submit.

The *D\_Household* data page fetches the entity details along with associated data, like Address.

## Updating an existing list property

Update the existing list properties for the household entity.

Fetch the household details to update the list of item details in a household that are captured as a part of Add Household. Update an existing list property, for example, HouseholdItemsList.

1. Update the *FetchHouseholdDetails (PegaPS-Data-Entity-Household)* data transform.

## 2. Add a step similar to AddressList.

The screenshot shows the 'Data Transform: Fetch household details [ Available]' configuration page. At the top, it displays the CL: PegaPS-Data-Entity-Household, ID: FetchHouseholdDetails, and RS: PegaPS:08-06-01. Below the header, there's a yellow banner stating 'This record has 1 info warning (including 1 unjustified)' with a 'View' link. The main area contains tabs for Definition, Parameters, Pages & Classes, Test cases, Specifications, and History. Under the 'Definition' tab, there is a table with three rows:

	Action	Target	Relation	Source
• 1	Comment	Set household details		
• 2	Apply Data Transf	SetLabel		
• 3	Set	.AddressList	equal to	D_AddressList[ID:pyID].p

### Result:

The Items list details are available for the user to update in the update case.

## Update Manage members case type

Update the existing members of a household case type. Using the *PegaPS-Work-Manage-Member* case type, you can update the details of the existing members of a household.

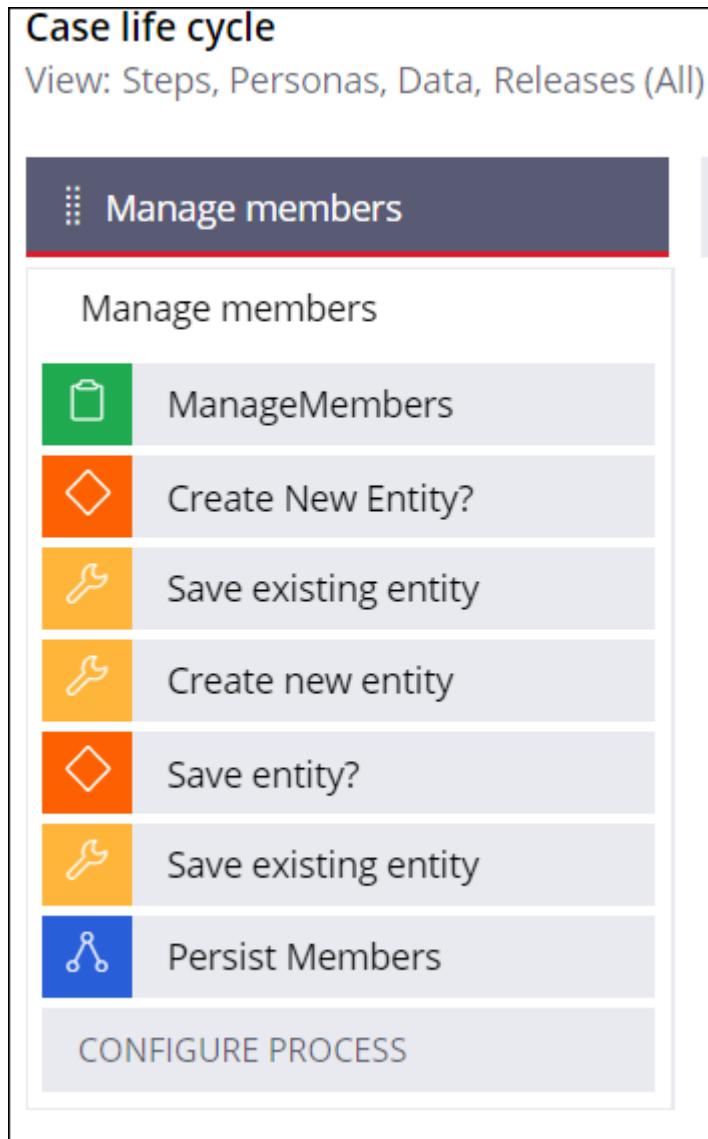
**Case life cycle**  
View: Steps, Personas, Data, Releases (All)

Manage members

Manage members

-  ManageMembers
-  Create New Entity?
-  Save existing entity
-  Create new entity
-  Save entity?
-  Save existing entity
-  Persist Members

CONFIGURE PROCESS



When you try to update existing members of a household by adding a new member or by updating the details of an existing member, the current household becomes inactive and a new household gets created with active status with the updated list of members.

**Rogers Family**

**Household members (3)**

- Mary Rogers (Jan 3, 2013 - Sep 1, 2021)
- Jack Rogers (Head of household • Jan 3, 2013 - Sep 1, 2021)
- Alex Rogers (Jan 3, 2013 - Sep 1, 2021)

Description: Rogers Family household

**Locations**

Map showing the location of 101 Tremont Street, Richmond, Virginia 23172 United States. Current address | Jan 3, 2013 - Present.

**Details**

Cases

## View Household case type

View the details of a household entity. Using the *PegaPS-Work-Entity-Household-View* temporary case type, you can view the complete details of an existing household entity.

**View**

**View**

**Fetch entity details**

**+ STEP**

The Details (*PegaPS-Data-Entity-Household*) wrapper section contains all the household related views or sections.

**Rogers Family**

**Household members (2)**

- Jack Rogers (Head of household) - Sep 1, 2021 - Present
- Alex Rogers (Sep 1, 2021 - Present)

Description: Rogers Family household

**Locations**

Map: 101 Tremont Street Richmond Virginia 23172 United States Current address | Jan 3, 2013 - Present

**Template**

1 Column with 100% width content

## Adding a field to the household view

Add new fields to an existing household view.

Add a field, for example, Is Remote location to the household view.

- To add the field, navigate to the Details wrapper section to show the field, or navigate to the Case details section.

**Rogers Family (HOUSEHOLD-1) ACTIVE**

**Household members (3)**

- Mary Rogers (Head of household) - Jan 3, 2013 - Present
- Jack Rogers (Jan 3, 2013 - Present)
- Alex Rogers (Jan 3, 2013 - Present)

**Associations**

**Template**

1 Column

Household type	Text
Dwelling type	Text
Dwelling type	Text
Household size	Text
Effective date	Date time
End date	Date time
Status	Text

- Add the new field along with the existing ones.

## Showing captured list details in the household view

Show the captured list, for example, HouseholdItemsList details in the household view.

1. Create a view or section in the *PegaPS-Work-Entity-Household-View* class.
2. Include the section in the Main content region of the *pyCaseMainInner* view as a tab.

The screenshot displays the Pega Government Platform interface. On the left, the 'Workarea' shows a 'Fixed summary pane' for the 'Rogers Family' household. It includes a 'Scorable summary pane' with details like ID (HOUSEHOLD-3), Type (Family), Dwelling type (Separate house), Size (2), Effective date (Sep 1, 2021), Household status (Active), Created by (PGP Manager), and Created on (1 year 7 months ago). Below this is a 'Details' section with a 'Cases' sub-section. In the center, the 'Main content' pane shows 'Household members (2)' with Jack Rogers (Head of household) and Alex Rogers listed. It also displays a 'Locations' section with a map of Chippenham Village showing the address 101 Tremont Street, Richmond, Virginia 23172, United States. On the right, the 'Case main inner' template editor is open, showing the 'Template' section with the 'Main case page UI template' selected. The sidebar lists various sections and their components, such as 'Fixed summary pane', 'Case Header', 'Case Action Header', 'Scorable summary pane', 'Case details', 'Main content (Grouped - Defau...', 'Case Information', 'My Cases List', 'Utils', 'Association utility', and 'Document'.

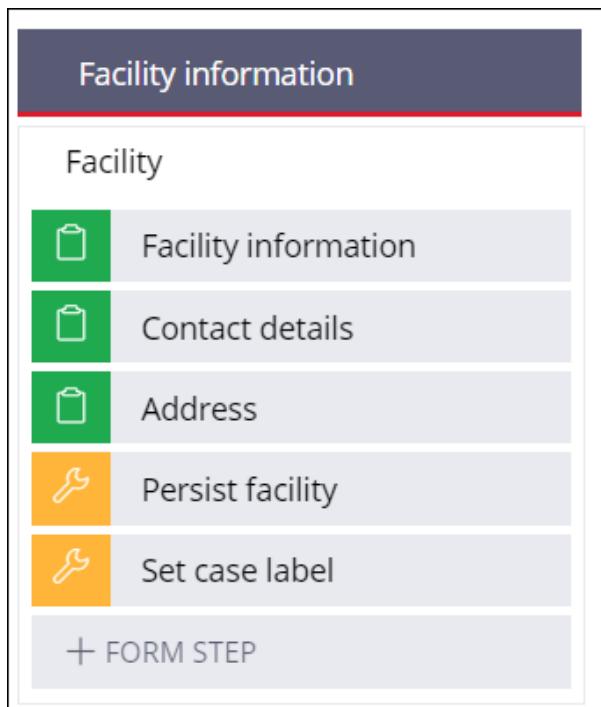
## Facility entity

Pega Government Platform provides data support to collect data about facilities in the form of Facility Entity.

The facility Entity feature contains a persistent datastore and three supporting case types to manage the Facility Entity datastore.

### Add Facility case type

Add a facility case type to the system. Using the *PegaPS-Work-Entity-Facility-Add* case type, you can introduce a new entity of the type Facility into the system.



After you add a new facility and successfully submit, the details of the facility are persisted along with supported data, for example, Address.



**Note:** Unlike other Pega Government Platform entities, facility does not have any duplicate validation check.

## Adding a new field to the facility intake screen

Capture a new field in the facility intake screen and persist it. You can add a new data field or property and extend the existing facility intake screen.

1. Create a field, for example, Permission Date in the facility data type.
2. Run the household case type and navigate to the Facility Information view.
3. Click the Add icon and add the field created in step 1.

The screenshot shows the Pega Government Platform Cosmos interface. On the left, there's a sidebar with 'Pulse' and 'Post' options. The main area displays a 'New Facility' form with three tabs: 'Facility information', 'Contact details', and 'Address'. The 'Facility information' tab is active, showing fields like 'Facility name\*', 'Category\*', 'Owner's name', 'Occupancy type\*', 'Facility status\*', 'Compliance status\*', 'Maximum occupancy', 'Current occupant', 'License status', and date pickers for 'Build start date' and 'Build complete date'. A 'Continue' button is at the bottom right. To the right of the form is a 'Facility information' template window titled 'Template' with a '3 Column (Inline Wrapping)' layout. It lists fields: 'Facility name' (Text input), 'Category' (Dropdown), 'Owner's name' (Text input), 'Occupancy type' (Radio buttons), 'Facility status' (Radio buttons), 'Compliance status' (Radio buttons), 'Maximum occupancy' (Number), 'Current occupant' (Text input), 'License status' (Dropdown), and 'Build start date' (Date time). Each field has its data type listed next to it.

### Result:

The new field automatically gets persisted in the household table.

## Introducing duplicate check validation for a field

Create duplicate check fields and update them with new fields.

1. Create the *CheckDuplicate (PegaPS-Data-Entity-Facility)* case match rule.
2. Add a new field by clicking Add must match condition, similar to how you add for Vehicle entity.

3. Update the filter logic accordingly.

Home CheckDuplicate

**Case Match: Check duplicate [ Available]**

CL: PegaPS-Data-Entity-Vehicle ID: CheckDuplicate RS: PegaPS:08-03-01

Evaluation Pages & Classes Specifications History

Case Matching Type  
Seek cases

**Must match conditions**

Conditions that must evaluate to true for case to be considered a duplicate

Label	Potential duplicates	Relationship	Current case
A	.VIN	is same	

+ Add must match condition

Hide filter logic A

4. Create an activity: *CheckDuplicate* (*PegaPS-Data-Entity-Facility*), and include the steps 2 and 3 by referring to the activity of the Vehicle entity.

The screenshot shows the configuration of the **Activity: Check duplicate [ Available ]**. The activity is defined under the class **PegaPS-Data-Entity-Vehicle** with ID **CheckDuplicate** and RS: **PegaPS:08-03-01**. The activity has two steps:

- Step 2:** A loop condition set to "When" followed by "Call SeekCases".
- Step 3:** A loop condition set to "When" followed by "Property-Set".

**Method Parameters**

Pass current parameter page

Name	Value
* Purpose	CheckDuplicate
* MatchingCasesPage	ResultList
CutoffScore	100

**Step 3 Method Parameters**

*PropertyName	*PropertyValue
.IsDuplicate	ResultList.pxResultCount > 0

5. Create a post processing activity for *FacilityOverview* (*PegaPS-Work*) and invoke the new `CheckDuplicate` activity as a first step in the activity with the step page as `.Facility` and in the second step, send a warning message when the property value `.Facility.IsDuplicate` is true.

Flow Action: FacilityOverview [Available]

Savable data pages

No items

[+ Add data page](#)

Apply cost

Apply data transform

Run activity

Back-to-back processing configuration

Look for an assignment to perform

If not found, look for assignments in other flows on this case

If not found, look for assignments in flows on the parent case

For each also consider assignments in work queues

If an assignment is not being performed

Show Harness

Confirm

## Adding a new list item

Update an existing list of items with new items.

Add a new list item, for example, InspectionList and persist it.

1. In Dev Studio, to add a new list item, create a data type Service in the PegaDATA database and have the following properties pyID (Primary key), EntityID, and EntityType.
2. Add other properties related to inspection like inspected by, inspected on, and so on.
3. In the Facility data type (*PegaPS-Data-Entity-Facility*), create a field.
  - a. In the Field name box, enter InspectionList.
  - b. In the Type list, select Embedded data.

- c. In the Data object list, select Inspection.
- d. Click Submit.

Add field to Facility

Field name *	InspectionList
Type	Embedded data
Data object *	Select...
Options	<input type="radio"/> Single record <input checked="" type="radio"/> List of records
> Advanced	

4. Update the *PersistEmbeddedPage (PegaPS-Data-Entity-Facility)* activity with a new step looping the InspectionList property and invoke the activity SaveObject by

passing the EntityID and EntityType parameters similar to CommList.

The screenshot shows the 'Activity: Persist embedded page [Available]' configuration screen. At the top, it displays the case level (CL: PegaPS-Data-Entity-Facility), activity ID (ID: PersistEmbeddedPage), and run set (RS: PegaPS:08-06-01). A yellow banner at the top right indicates 'This record has 1 info warning (including 1 unjustified)'. Below the banner, there are tabs for Steps, Parameters, Pages & Classes, Security, Test cases, Specifications, and History. The Steps tab is selected, showing two steps:

- Step 1: Label (empty), Method: Call LoadAddressListLatLong, Step page: (empty)
- Step 2: Label (empty), Method: Call SaveObject, Step page: .CommList

Under 'Method Parameters', there is a checkbox 'Pass current parameter page' which is unchecked. Below it, a table lists parameters with their names and values:

Name	Value
* EntityID	Primary.pyID
* EntityType	Primary.EntityType

5. Create a UI to capture the facility details so that they persist after you enter the data.

## Update Facility case type

Update the existing facility case type. Using the *PegaPS-Work-Entity-Facility-Update* case type, you can update the details of an existing entity.

Update facility

Facility

-  Facility information
-  Contact details
-  Address
-  Persist facility
-  Set case label

+ FORM STEP

The Update facility case type fetches the details of the Facility entity that needs to be updated. You can make changes to the facility data and submit them. The details persist after you submit.

The *D\_Facility* data page fetches the entity details along with associated data, like Address, Communication, and so on.

## Updating an existing list property

Update an existing list property, for example, *InspectionList*.

Fetch the facility details to update the list of item details in a facility that are captured as a part of Add Facility.

1. Update the *FetchFacilityDetails (PegaPS-Data-Entity-Facility)* data transform.

## 2. Add a step similar to AddressList.

Action	Target	Relation	Source
• 1	Comment	Set Facility details	
• 2	When .pyID=""		
• 2.1	Exit Data Transfo		
• 3	Apply Data Trans v	SetObjClass	
• 4	Set .AddressList	(with a gear icon) equal to	D_AddressList[ID:Param] (with a gear icon)
• 5	Set .CommList	(with a gear icon) equal to	D_CommunicationList[Id] (with a gear icon)

## View Facility case type

View the details of a facility entity. Using the *PegaPS-Work-Entity-Facility-View* temporary case type, you can view the complete details of an existing facility entity.

The Details (*PegaPS-Data-Entity-Facility*) wrapper section contains all the facility related views or sections.

**BlueSky Storage**

**Details**

Name	BlueSky Storage	Compliance status	Compliant
Category	Storage	Occupancy	---
Owner	Marty Solomon	Current occupant	---
Non profit	No	Occupancy type	Owned
Facility status	Active	Build complete date	Jul 31, 2017
Build start date	Jan 2, 2017	License status	Up to date
Last inspection	Oct 31, 2019	Facility description Bluesky storage facility provides solution for storage of personal and industrial goods.	

**Locations**

Map Satellite

222, South Hermitage Avenue  
Chicago Illinois 60605 United States  
Primary  
Mapped

**Template**

1 Column with 100% width content

**Settings**

## Adding a field to the facility view

Add new fields to the existing facility view.

Add a field, for example, Permission Date to the facility view.

1. To add the field, navigate to the Details wrapper section and choose the appropriate section to show the field, for example, the Facility header RO section.

The screenshot shows a facility view for 'BlueSky Storage'. The main content area displays details such as Name (BlueSky Storage), Category (Storage), Owner (Marty Solomon), Non profit (No), Facility status (Active), Build start date (Jan 2, 2017), Last inspection (Oct 31, 2019), Compliance status (Compliant), Occupancy (Occupant), Current occupant (Marty Solomon), Occupancy type (Owned), and Build complete date (Jul 31, 2017). Below these details is a Facility description: 'Bluesky storage facility provides solution for storage of personal and industrial goods.' The Locations section shows a map with a pin labeled 'Plann Engle' at 222, South Hermitage Avenue, Chicago Illinois 60605 United States Primary. A 'Mapped' status is also indicated. On the right, a template editor panel titled 'Template A' shows a single column with 100% width content.

2. Add the new field along with the existing ones.

## Showing captured list details in the facility view

Show the captured list, for example, InspectionList details in the facility view.

1. Create a view or section in *PegaPS-Work-Entity-Facility-View* class.
2. Include the section in the Main content region of the *pyCaseMainInner* view.

The screenshot shows a facility view for 'BlueSky Storage' with a scrollable summary pane. The main content area displays details such as Name (BlueSky Storage), Category (Storage), Owner (Marty Solomon), Non profit (No), Facility status (Active), Build start date (Jan 2, 2017), Last inspection (Oct 31, 2019), Compliance status (Compliant), Occupancy (Occupant), Current occupant (Marty Solomon), Occupancy type (Owned), and Build complete date (Jul 31, 2017). Below these details is a Facility description: 'Bluesky storage facility provides solution for storage of personal and industrial goods.' The Locations section shows a map with a pin labeled 'Plann Engle' at 222, South Hermitage Avenue, Chicago Illinois 60605 United States Primary. A 'Mapped' status is also indicated. On the right, a template editor panel titled 'Template' shows the 'Main content' section with various components: Fixed summary pane, Case Header, Case Action Header, Scrollable summary pane, Case details, Main content (Grouped - Defau, Case Information, My Cases List, and Utils (Association utility, Document).

## Search entities

Search for information across multiple record types. The search entities feature returns real-time results in entity data.

You can directly access the search page of an entity from the navigation panel. Each entity has its own landing page for search. You can build a landing page for an entity using the associated harness.

The following table shows the list of harnesses for each entity:

Entity	Harness class	Harness name
Person	PegaPS-Data-Portal	Person
Business	PegaPS-Data-Portal	Business
Facility	PegaPS-Data-Portal	Facility
Vehicle	PegaPS-Data-Portal	Vehicle
Household	PegaPS-Data-Portal	Household

The following table shows a list of data pages and report definitions that you can use for each entity search from the landing pages.

Entity	Data page	Report definition
Person	D_PersonList	PersonList
Business	D_BusinessList	BusinessList
Facility	D_FacilityList	FacilityList
Vehicle	D_VehicleList	VehicleList
Household	D_SearchHouseholdList	SearchHouseholdList

Each harness contains two areas, the header area and the content area. The Content region holds the section which shows the entities in a grid. The following are the sections for each entity:

Entity	Section Class	Section name
Person	PegaPS	SearchPersonTable
Business	PegaPS	SearchBusinessTable
Facility	PegaPS	SearchFacilityTable
Vehicle	PegaPS	SearchVehicleTable
Household	PegaPS	SearchHouseholdTable

## Introducing new search criteria

Complete the following steps to introduce new search criteria for an entity:

1. Open the data page and the report definition of the entity.
2. Add the new search criteria as a parameter.
3. Update the filer logic of the report definition with the appropriate relationship.
4. Add the property in the **Edit columns** list of the report definition.
5. In the section, pass the search text to the newly added parameter of the data page. If you cannot view the parameter, reselect data page.
6. Add the property as a column to the table in the section.

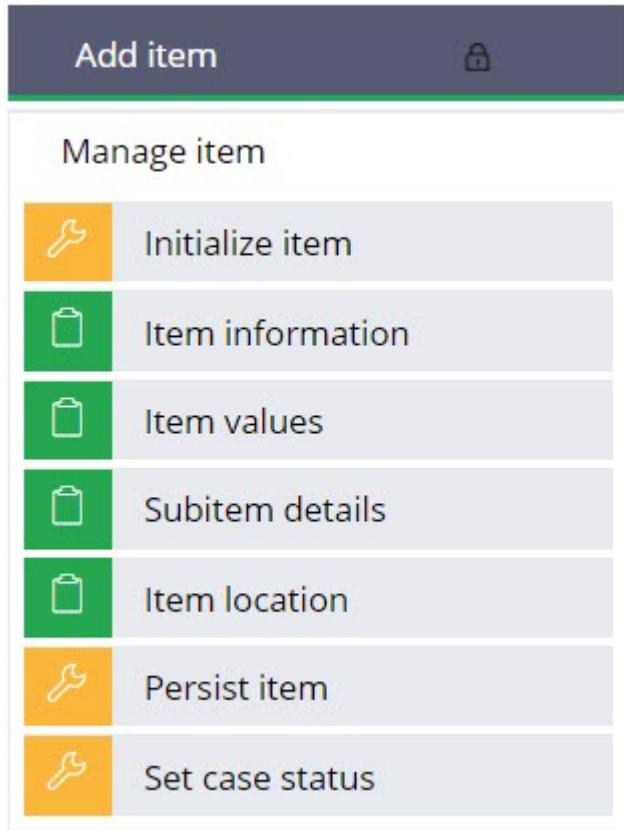
## Item entity

Pega Government Platform provides data support to collect data about item in the form of Item Entity.

The Item Entity feature contains a persistent datastore and three supporting case types to manage the Item Entity datastore.

### Add Item case type

Add an item case type. Using the *PegaPS-Work-Entity-Item-Add* case type, you can introduce a new entity of the type Item into the system.

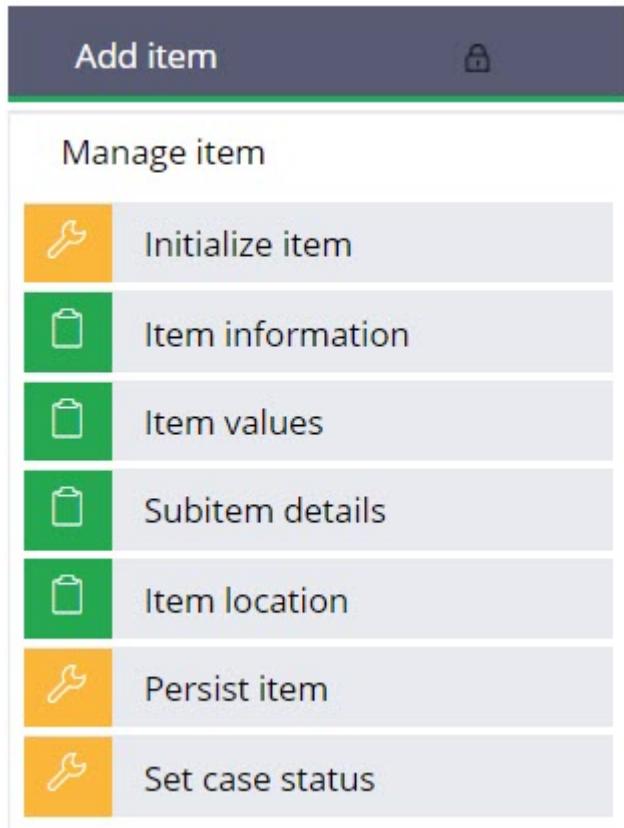


#### *Add item case type*

The Add Item case type checks for duplicate item entries and notifies about the same. The case type persists the item, record along with the supported data like item values, subitem details, and item location.

### **Adding a new field to the item intake screen**

Capture a new field in the item intake screen. You can add a new data field or property and extend the existing item intake screen.



*Add new field in item case type*

1. In the Item data type, create a field. For example, Short name.
2. Run the Item case type, and then navigate to the ItemInfo view.
3. Click the Add icon of Short name, and then you can see the field.

The screenshot shows the 'Add new field - item intake form' screen. On the left, there is a form with fields for Physical description (Size, Color, Fabric, Style), Description, and Short name. On the right, there is a sidebar with various search and update operations listed:

- Search business
- SearchFacilities
- SearchPerson
- Select or create
- selection
- Short name**
- Status
- Style value
- Sub category**
- Subject count
- Subject inskey
- Update Operator
- Update Operator Name
- Update System ID

Add new field - item intake form

### Result:

The new field is automatically saved in the item table.

## Updating duplicate check validation for a field

Update the existing duplicate check fields with new fields.

1. Open the *CheckDuplicate (PegaPS-Data-Identifier)* case match rule.
2. Add a new field by clicking Add must match condition.
3. Update the filter logic accordingly.

**Case Match: Check duplicate [Available]**

CL: PegaPS-Data-Identifier ✓ ID: CheckDuplicate RS: PGPCosmos:08-08-01

Evaluation Pages & Classes Specifications History

Case Matching Type  
Seek cases

**Must match conditions**

Conditions that must evaluate to true for case to be considered a duplicate

Potential duplicates	Relationship	Current case
.IDNumber	is same	
.IDType	is same	

+ Add must match condition  
Show filter logic

#### Duplicate check validation

4. Click Save as.
5. In the Context section, select an application.
6. In the Add to ruleset area, select a ruleset and the ruleset version.
7. Click Create and open.

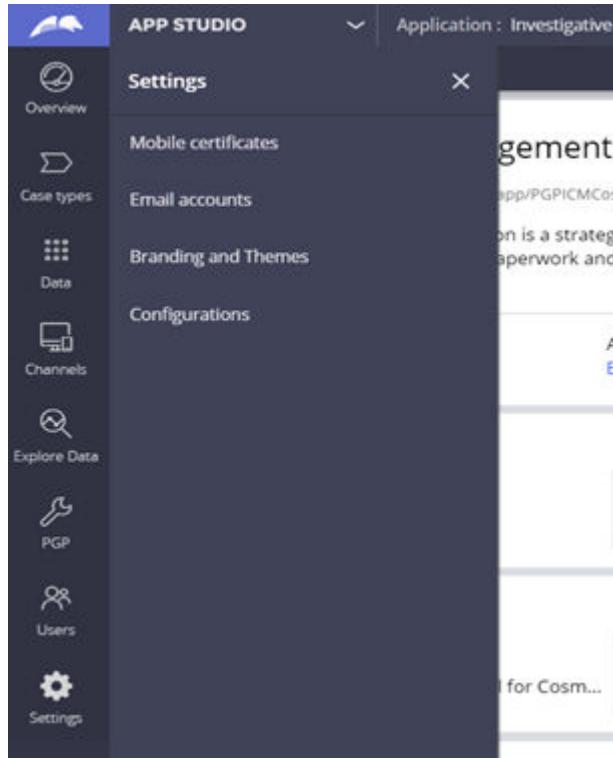
## Enable/Disable Duplicate check

Duplicate check for the item is always enabled. You can enable/disable the duplicate check from the configuration in App studio. If duplicate check is disabled, On creation of new item, duplicate check will not be happened and vice versa.

Follow these steps to enable or disable duplicate check:

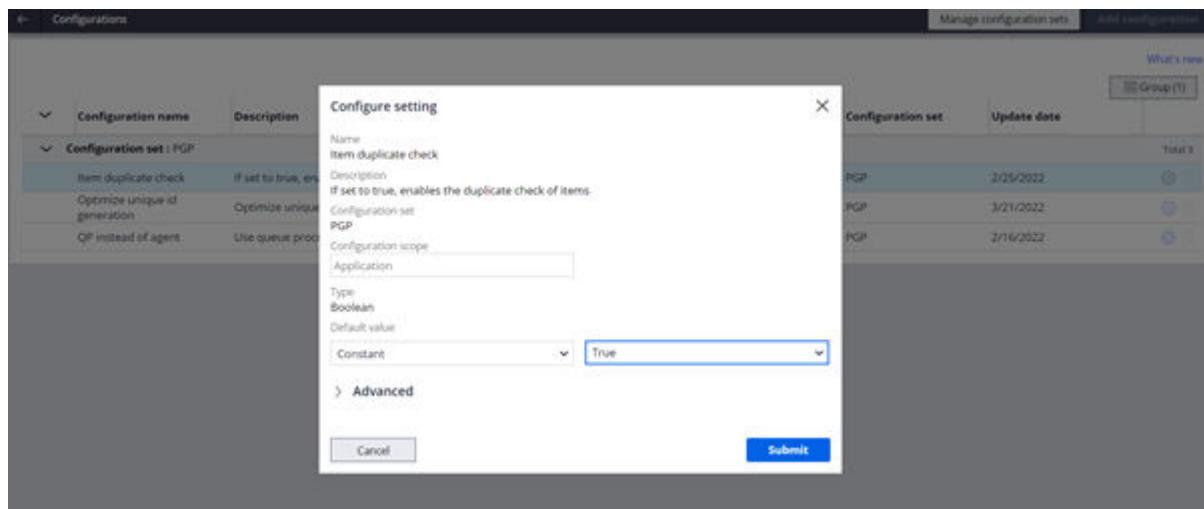
1. Login to Pega Government Platform with admin credentials and switch to App studio.

2. From the left pane, click Settings > Configuration.



*Enable/disable duplicate check*

3. Click gear icon of duplicate check row.
4. Change the value of constant dropdown. Select True to enable the duplicate check else False to disable the duplicate check.



*Configuration setting to enable/disable duplicate check*

## Adding a new list item

Update an existing list of items with new items.

Add a new list item, for example, SellerList and persist it.

1. In Dev Studio, in the PegaDATA database, create the SellerList data type with the following properties: *pyID* (Primary key), *EntityID*, and *EntityType*.
2. Add other properties that are related to the SellerList, such as the Seller name, description, and so on.
3. In the Item data type (*PegaPS-Data-Entity-Item*), create a field of type page list of SellerList data type class.
4. Update the *PersistEmbeddedPage* (*PegaPS-Data-Entity-Item*) activity with a new step that loops the *SellerList* property, and then invoke the activity *SaveObject* by passing the *EntityID* and *EntityType* parameters similar to *IdentifierList*.

Activity: Persist embedded page [Available]

PegaPS-Data-Entity-Item ID: PersistEmbeddedPage RS: PGPCosmos:08-08-01

This record has 1 severe or moderate warning and 1 info warning [View](#)

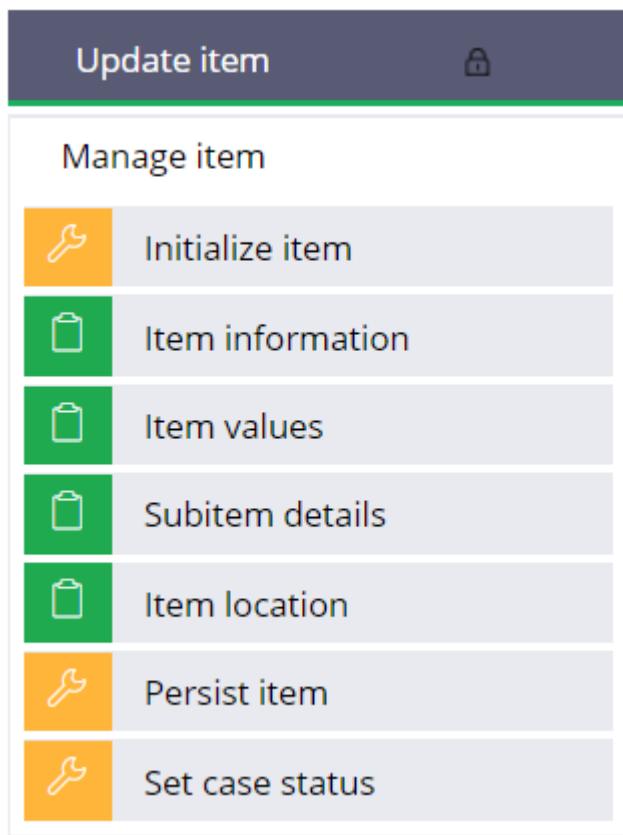
Steps	Parameters	Pages & Classes	Security	Test cases	Specifications	History
1.	Loop When >	Call LoadAddressListLatLong				Invoke activity to load GeoCodeAPI for address lat long
2.	Loop When >	Call RemoveLink				Deleting Subitems
3.	Loop When >	Call RemoveLink				Deleting Manufacturers
4.	Loop When >	Call SaveObject				Save ID List
5.	Loop When >	Call SaveObject				Save image
6.	Loop When >					TempDeleteValueList.pxRes Deleting valueList
1.	Loop When >	Call RemoveLink				Remove Assignee Link
2.	Loop When >	Call RemoveObject				Remove Value
7.	Loop When >	Call SaveObject				To save valuesList
8.	Loop When >	Call SaveLink				To save Items Associations
9.	Loop When >	Property-Set				To set ClassTo of Owner
10.	Loop When >	Call SaveLink				Saving owner

*Adding a new list item*

5. Create a UI to capture the SellerList details.

## Update Item case type

Update the existing *Item* case type. Using the *PegaPS-Work-Entity-Item-Update* case type, you can update the details of an existing entity.



#### *Update item case type*

You can make changes to the item data and submit them. The details persist after the duplicate validation check ensures that there are no duplicates.

The *D\_Item* data page fetches the entity details along with associated data like item values, subitems, and so on.

### **Updating an existing list property**

Update an existing list property, for example, SellerList .

You can update the existing list properties for the item entity. Fetch the SellerList details to update the existing list of SellerList that are captured as a part of the Add Item procedure.

1. Update the *FetchItemDetails* (*PegaPS-Data-Entity-Item*) data transform.
2. Add a step similar to ValuesList or AddressList to set the details using the corresponding SellerList data page of type: List.

**Data Transform: Fetch item details [Available]**

PegaPS-Data-Entity-Item → FetchItemDetails PGCosmos:08-08-01

This record has 1 info warning (including 1 unjustified) [View](#)

	Action	Target	Relation	Source
• 1	Comment	Load item details		
• 2	Set	.AttachmentRefID	equal to	@toUpperCase(pxObjClass) Select values +
• 3	Apply Data Transf	SetObjClass		
• 4	Apply Data Transf	PrepareObject		
• 5	Apply Data Transf	SetLabel		
• 6	Set	.ObjectType	equal to	.EntityType Select values +
• 7	Set	.RefreshAssociationKey	equal to	.EntityID+“Association” Select values +
• 8	Set	.RefreshLocationKey	equal to	.EntityID+“Location” Select values +
• 9	Set	Param.PoolID	equal to	“ItemMore” Select values +
• 10	Set	Param.a	equal to	@Utilities.pxExecuteAnActiv Select values +
• 11	Set	.AddressList	equal to	D_AddressList[ID:Param.ID] Select values +
• 12	Set	.ValuesList	equal to	D_ValueList[ID:Param.ID].p Select values +
• 13	Set	.IdentifiersList	equal to	D_IdentifiersListBySequence Select values +
• 14	Set	.SubItemsList	equal to	D_SubItemsList[ID:Param.ID] Select values +

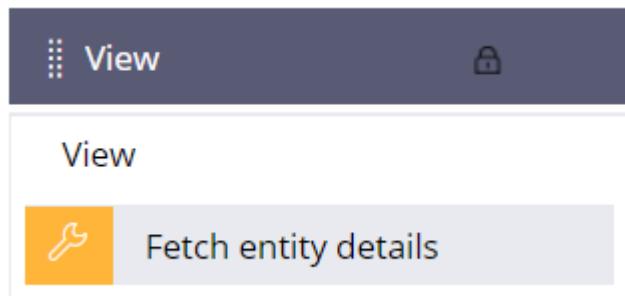
*Updating a existing list property for Item entity*

### Result:

The Update case populates the details and any changes made are persisted.

## View Item case type

View the details of a item entity. Using the *PegaPS-Work-Entity-Item-View* case type, you can view the complete details of an existing item entity.



*View item case type*

The *EntityDetails (PegaPS-Data-Entity-Item)* wrapper section contains all the item related views or sections.

This screenshot shows the EntityDetails (PegaPS-Data-Entity-Item) wrapper section. It includes several sub-sections:

- Details:** A table showing item details such as Name (Dinning table), Category (Household goods), Condition (Good), and Description (test).
- Subitems:** A table listing sub-items with columns for Item ID, Name, Category, and Serial number. It includes entries for ITEM-1 (Morgan Consolidated 47 AK) and ITEM-1002 (Gun).
- Manufacturers:** A table showing manufacturer details with columns for ID, Legal name, Trade name, and Ownership type. It includes an entry for PEG.

*View item case type*

## Adding a field to the item view

Add new fields to an existing item view.

Add a field, for example, Short name to the item view.

- To add the field, navigate to the *EntityDetails* wrapper section and choose the appropriate section to show the field, for example, the Entity details section.

*Add field in item view*

- Add the new field along with the existing ones.

## Showing captured list details in the item view

Show the captured list, for example, SellerList details in the item view.

- Create a view or section in the *PegaPS-Work-Entity-Item-View* class.
- Include the section in the Main Content region of the *pyCaseMainInnner* view.

The screenshot displays three distinct panels within the Pega interface:

- Left Panel (Details View):** Shows a summary of a 'Dinning table' item with ID ITEM-1007. It includes fields like Name (Dinning table), Category (Household goods), Condition (Good), and Subitems (2). A 'Scorable summary pane' is also present.
- Middle Panel (Template Configuration):** Displays the 'Main case page UI template'. It includes sections for 'Case Header', 'Case Action Header', 'Case details', and 'Main content (Grouped - Default)'. Under 'Main content', there are entries for 'Case Information', 'Programs tab', 'My Cases List', and 'Entity History tab'.
- Right Panel (Configuration):** Shows the configuration for a 'Scorable summary pane'. It lists various components such as 'Case Header', 'Case Action Header', 'Case details', and 'Main content (Grouped - Default)'.

*Showing captured list details*

## Copy item to a new case type

If you want to create a new item which is similar to the existing item with some minor changes, then instead of creating a new item and filling up all the details, in a single click you can copy the details to the new case.

Click Action > Copy from the item screen which you want to copy. A new item case screen pops up with the item details from which you performed the action.

## Entity attachments

Attach documents to the entity profile. You can view, download, update, or delete these documents at anytime.

In government applications, various entities hold different documents which you can view or refer in other cases. For example, a Person entity can have a passport, a birth certificate, and so on, attached to their profile. If required, you can refer these attachments in the license case.

The following are some of the features of entity attachments:

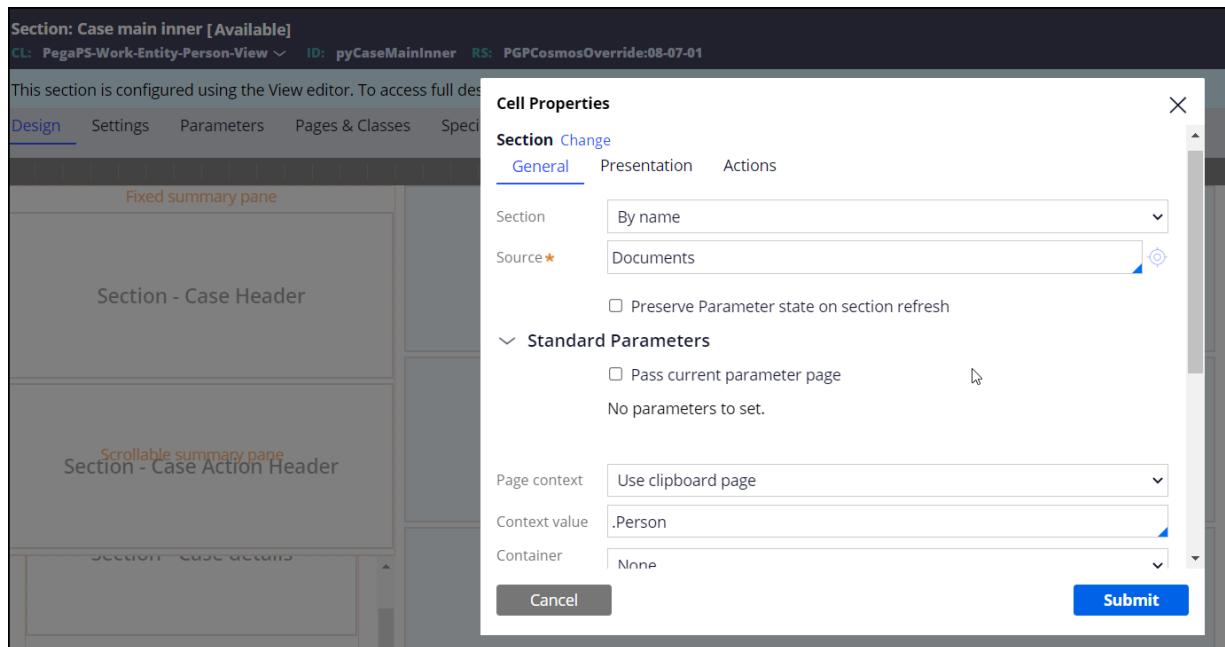
- Entity attachments offers support for entities like person, business, facility, vehicle, and household.
- When you add a new attachment, PGP creates a new document case (*CL: PegaSocial-Document*).
- The *AttachmentRefID* property of the entity is set as the parent instance handle key of the document case. Using *AttachmentRefID*, the entity view screen displays the attachments of an entity.

## Extending attachments for a new entity type

Hold documents for the entities of a new entity type in the implementation layer.

1. In **pyCaseMainInner** section of entity view class, in the **Utils** area, add a **Documents** section with the Page context as Use clipboard page and context value as Entity Page.

For example: For Person entity, in the **pyCaseMainInner** section (*CL:PegaPS-Work-Entity-Person-View*), in the Utils region, add a Documents section with Page context as Use clipboard page and context value as **.Person**.



2. Before persisting the entity, set the *AttachmentRefID* property. Set this property in step 3 of the *PrepareObject* (*CL:PegaPS-Data-Entity*) data transform.

**Note:** If this setting is not getting called for your new entity, then before persisting, set the *AttachmentRefID* property to  
 '@toUpperCase(.pxObjClass)+" "+.EntityID'

3. Similarly, during the fetching of entity details, set the *AttachmentRefID*. In the data transform of the new entity's lookup datapage, set *AttachmentRefID* to '@toUpperCase(.pxObjClass)+" "+.EntityID'. Refer *FetchPersonDetails* (*CL:PegaPS-Data-Entity-Person*) step 2.

## Entity History and Field Audit

Pega Government Platform provides support for History and Field auditing for necessary fields, for entities.

The following are some of the features of entity history and field audit:

- The history of an entity shows details such as by whom and when it is updated/added and the case where this action is performed.
- Field Audit of an entity shows previous values and current values of a property along with time and user.

For example, the following images show the history and field audit of a person entity.

The screenshot displays two panels side-by-side. The left panel is titled 'Jame Bonds' and shows a list of personal details for this entity. The right panel is titled 'HISTORY FIELD AUDIT' and displays a table of audit logs.

**Entity Details (Left Panel):**

ID	PERSON-23910
Home phone	(546) 745-6745
Mobile phone	(546) 745-6745
Email	bondjamebond007@gmail.com
Home	7-4-42/2 hyd Delhi 256254 India
Mailing	Same as above
Contact preference	Mail
Contact time	Any
Language	English
Created by	Ankireddy, Gurusai Ravi Raja Reddy
Created	6 minutes ago

**Entity History (Right Panel):**

Time	Description	Performed by
03/29/2022 04:56 PM	Updated Person details from A-38006	ankig
03/29/2022 04:50 PM	Updated Person details from UP-23004	ankig
03/29/2022 04:49 PM	Updated Person details from UP-23003	ankig
03/29/2022 04:46 PM	Added Person details from AP-32005	ankig

*Entity history*

Time	Description	Performed by
03/29/2022 04:50 PM	Changed Last name From 'Bond' To 'Bonds'	ankig
03/29/2022 04:50 PM	Changed First name From 'James' To 'Jame'	ankig
03/29/2022 04:46 PM	Added Last name 'Bond'	ankig
03/29/2022 04:46 PM	Added First name 'James'	ankig

### Field audit

- Extend History and Field audit to a new entity
- Extend field audit to a new field of an existing entity

## Extend History and Field audit to a new entity

To add history to a new entity open `pyCaseMainInner` of entity view class and add Entity History section with the Page context as Use clipboard page and context value as Entity Page.



**Note:** Use `SaveEntity(CL:PegaPS-Work)` to Persist the entity.

The next step is to create a report definition for history in entity history-data class, save as `PersonHistory(CL:History-PegaPS-Data-Entity-Person)`, and add the report to **D\_EntityHistory** data page as a source.

Similarly, create a report definition for field audit in entity history-data class, save as `PersonFieldAudit (CL:History-PegaPS-Data-Entity-Person)`, and add the report to **D\_EntityFieldAudit** data page as a source.

For field auditing, create a declare trigger on data-class of the entity, set Trigger when an instance is dropdown to Saved. Under Trigger activity, set Name to TrackSecurityChanges and set Execute dropdown to immediately. And create **pyTrackSecurityChanges** data transform with data-class of the entity and add fields to audit.

## Extend field audit to a new field of an existing entity

For example, to audit a new field, say **.EyeColor**, for Person Entity open *pyTrackSecurityChanges* (CL:PegaPS-Data-Entity-Person) data transform and add **.EyeColor** in target, add label for EyeColor in source.

To audit page list property fields like country in Addresslist (Page list Property), follow the image.

Action	Target	Relation	Source
Set	.Name.FirstName	equal to	First name
Set	.Name.LastName	equal to	Last name
Set	.AddressList(1).pyCountry	equal to	Country
Set	.AddressList(1).pyState	equal to	State
Set	.AddressList(1).pyCity	equal to	City
Set	.AddressList(1).AddressLine1	equal to	AddressLine1
Set	.AddressList(1).AddressLine2	equal to	AddressLine2
Set	.AddressList(1).PostalCode	equal to	Postal code

Field audit

# Entity merge

## PGP Merge Entities

The PGP Merge Entities is a feature for merging 2 or more entities into a single entity which acts as a master.

## Architecture

Merge entities consists of two parts:

- Configuring Scalar and List merge attributes (Properties) in the design time.
- Executing the merge for the selected entities at run time.

Merge attributes can be configured from App Studio by navigating to the PGP > Merge entities landing page.

Scalar attributes can be configured by providing the following details:

- Name of the property
- Property supported for merge and
- Data type of the property

List attributes can be configured by providing the following details:

- Name of the attribute
- Data page that fetches the list items and
- Properties to be shown to the user

## Merge Entities

In the Merge entities case type, User can select only one of the values for scalar attributes from of the selected entities to be the value of the master entity record.

For list attributes, Users can select any record to be part of the master entity without any restriction.

## Technical details for merging entities

To support merging of an entity, 2 properties are added to the entity data model and corresponding columns are also added to the respective entity database table.

- MergeStatus (Possible values: Active/Inactive)
- MergedTo

For example, let's assume below are the entities before merge

Entity ID	Merge Status	Merged To
PERSON-1	--	--
PERSON-2	--	--

Now user selects PERSON-1 as a master entity and wants to merge PERSON-2 into PERSON-1, up on completion, the entities will have the following data to the above mentioned two properties.

Entity ID	Merge Status	Merged To
PERSON-1	Active	--
PERSON-2	Inactive	PERSON-1

Along with the above values to support the unmerge functionality, the Master entity would have the details that it possesses before the merge in the Snapshot (**PegaPS-Data-Snapshot**) embedded page of the entity.

## Technical details for merging list items

The design of merging the list items like Address has a different way of handling the merge compared to merging of entities.

For handling the merge of list items 2 properties are introduced to the data model.

- MergeStatus (Possible values: MasterInactive/Inactive/Active)
  - MasterInactive: Used to deactivate the master list items (only if not selected), these are omitted and not shown in master profile.
  - Inactive: Used to deactivate the duplicate list items (irrespective of selection is made or not.)
  - Active: Used to activate the list items that are selected from the master items.
- RefID

For example, let us assume below are the addresses of the entities in the system.

<b>Address ID</b>	<b>Entity ID</b>	<b>Merge Status</b>	<b>RefID</b>	<b>User selection</b>
Address-1	PERSON-1 (Master)	--	--	Selected by user
Address-2	PERSON-1 (Master)	--	--	Not selected
Address-3	PERSON-2 (Duplicate)	--	--	Selected by user
Address-4	PERSON-2 (Duplicate)	--	--	Not selected

In the above scenario, one address from each entity is selected by the user and below are the address details after the merge.

<b>Address ID</b>	<b>Entity ID</b>	<b>Merge Status</b>	<b>RefID</b>
Address-1	PERSON-1 (Master)	Active	--
Address-2	PERSON-1 (Master)	MasterInactive	--

Address ID	Entity ID	Merge Status	RefID
Address-3	PERSON-2 (Duplicate)	Inactive	--
Address-4	PERSON-2 (Duplicate)	Inactive	--
Address-5	PERSON-1 (Master)	Active	Address-3

Below is the explanation for each address how its handled in the above scenario.

1. Address-1: This originally belongs to master entity and it has been selected by the user, so the status is set to Active.
2. Address-2: This originally belongs to master entity and it has not been selected by the user, so the status is set to MasterInactive (this is not shown in master profile going forward)
3. Address-3: This originally belongs to duplicate entity, so the status is set to Inactive.
4. Address-4: This originally belongs to duplicate entity, so the status is set to Inactive.
5. Address-5: This is a new address added to master by cloning the **Address-3** which is a selected address from the duplicate entity and the status is set to Active. The RefID property is set to the value Address-3 to indicate that it is cloned from Address-3.

The key concept is all the selected list items from the duplicate are cloned with new ID's and added to the master.

## Creating New Entities

## What are PGP entities?

PGP Entities are a collection of robust data structures that model data elements commonly used in government processes. The PGP Entities are persistent data classes that come pre-configured as a part of PGP.

- Person
- Business
- Vehicle
- Facility
- Household

Entity data structures serve two distinct purposes within PGP:

1. Entity instances are persisted to a data store like any local data type or data table
2. Entity data classes are used as a type for embedded pages defined on the PGP work ancestor to support case processing

## Persistent Data Store

An entity data class (Data- class) is managed (curated) in an independent data store (database table) by the *Add, Update, View and Search* case types (Work- classes).

## Case Type Embedded Page

An entity data class is also used as a type for embedded pages incorporated into the PGP work ancestor. These embedded pages are used by the PGP component processes to implement PGP features.

PGP Entities follow a pattern of Persistent Data curated by Business Processes. This pattern allows for the possibility that in a customer implementation the entity datastore may be mapped to customer resource that is external to Pega and PGP. In this pattern a persistent datastore descended from *PegaPS-Data-Entity* is added, updated, viewed, and searched by Case Types descended from *PegaPS-Work-Entity*.

## New Entity Overview

This guide will step through the process of extending the PGP Entity data structure adding a new entity called *Item* built in a PGP implementation layer.

In the following examples *PGPImp* is the organization name of the implementation application and *DemoPGP* is the name of the implementation application. The class *PGPImp-DemoPGP-Work* inherits (directed) from the *PegaPS-Work* class.

## Create the Persistent Datastore

*PGPImp-DemoPGP-Data-Entity-Item*

- Properties relevant to the entity
- View(s) for editing, reviewing
- Data Pages for retrieving
- Optional: Case match rule to avoid duplicates, and validation rules for data

## Extend the Work Class Properties

In the primary work class of the implementation application, *PGPImp-DemoPGP-Work*, add a page property of type *PGPImp-DemoPGP-Data-Entity-Item* to hold *Item* data during case management processing.

## Create the Data Management Case Types

- *PGPImp-DemoPGP-Work-Entity-Item-Add*
- *PGPImp-DemoPGP-Work-Entity-Item-View*
- *PGPImp-DemoPGP-Work-Entity-Item-Search*
- *PGPImp-DemoPGP-Work-Entity-Item-Update*

## Enable for App Studio

- Use views and design templates to construct the user interface
- Catalog the new Entity's rules in Relevant Records so they will appear in the App Studio smart prompts

- Configure whether case types will appear on +Create nav

## Development Steps

While it is possible to do some of this work in App Studio, the approach discussed here is based on a Dev Studio experience. App Studio hides some development details, like class names and rulesets, in order to present a development environment that can focus on case types and business processes, but for this exercise we need full control over those details.

## Create Persistent Datastore

Create a persistent data type and supporting rules for the new entity.

### Create Data Type Ancestor

Add an abstract data class to the implementation layer to leverage the Enterprise Class Structure pattern and provide extension points for reusable rules common to the *Entity* implementation layer data types.

1. Use Create > SysAdmin > Class to create a new class
  - a. For the *Label* use *Entity*
  - b. For the *Class Name* use *PGPImp-DemoPGP-Data-Entity*
  - c. Create and open
  - d. For *Select class type* use *Abstract*
  - e. For *Created in version* use the current version
  - f. For *Parent class (directed)* use *PegaPS-Data-Entity*
2. Save the rule.

### Create Data Type

1. Use Data types > Data types > Add data type wizard to create the new *Item* data type and supporting rules.
  - a. Set the *Label as Item*
  - b. Add an appropriate *Description* for the class

- c. Under Advanced set the parent class to *PGPImp-DemoPGP-Data-Entity*
  - d. *Submit* to create the new *Item* data type
2. Configure the *Item* data type
- a. Use the *Data model* tab, add properties to the *Item* data type
    - 1. *EntityID* of type *Text*
    - 2. *pyID* of type *Text*
    - 3. Add additional properties as needed
  - b. Under the *Sources* tab *Create a local source*
  - c. Select *pyID* as the Key (*Use as key*)
  - d. *Submit* to create a local source for the *Item* data type

## Create View UI for the New Data Type

Build the basic data layer UI rules to present the data of the new *Item* data type.

- 1. Create View *ItemDetail* with Applies-To of *PGPImp-DemoPGP-Data-Entity-Item*
  - a. Place all the relevant properties from the *Item* data type
- 2. Create View *ItemDetailRO* with Applies-To of *PGPImp-DemoPGP-Data-Entity-Item*
  - a. Place all the relevant properties from the *Item* data type, with the *Presentation -> Edit Options* set to *Read-Only*.

## Extend the Work Ancestor Class

Add a Page property to the implementation layer work ancestor of type *PGPImp-DemoPGP-Data-Entity-Item* to hold an *Item* instance during case management processes.

Create a new Page property *.Item* of type *PGPImp-DemoPGP-Data-Entity-Item* with Applies-To *PGPImp-DemoPGP-Work*.

- 1. Use *Create > Data Model > Property* to add a new property
  - a. For the *Label* use *Item*
  - b. For the *Apply to* use *PGPImp-DemoPGP-Work*
  - c. Create and open
  - d. Change the Property type to Single Page
  - e. Select *PGPImp-DemoPGP-Data-Entity-Item* as the *Page definition*

f. Save the rule.

## Review Entity Data Type Rules

Switch to App Explorer (App) and show *PGPImp-DemoPGP-Data-Entity-Item*. Review all the rules created for the *Item* data entity.

Applies -to	Rule Name	Rule Type	Description	Comment
	PGPImp-DemoPGP-Data-Entity	Class	Abstract	
	Item		Data Type	
	PGPImp-DemoPGP-Data-Entity-Item	Class	Concrete	Key pyID
	PGPImp-DemoPGP-Data-Entity-Item	Database Table	Persistence	
PGPImp-DemoPGP-Data-Entity-Item	ItemName	Property	Text	
PGPImp-DemoPGP-Data-Entity-Item	... more properties	Property	Text	
	D_Item	Data Page	Page	Source=Lookup

Applies -to	Rule Name	Rule Type	Description	Comment
	D_ItemList	Data Page	Page List	Source=Report Definition
	D_ItemSaveable	Data Page	Page	
PGPImp-DemoPGP-Data-Entity-Item	DataTableEditorReport	Report Definition		All Item Properties

## Create Case Types

## Create Abstract Support Work Classes

When a new implementation layer is generated by the New Application Wizard it only creates descendants for the selected concrete classes. Manually add the missing abstract work classes to leverage the Enterprise Class Structure pattern and provide extension points for reusable rules common to the *Entity* and *Item* implementation layer work types.

1. Use Create > SysAdmin > Class to create a new class.
  - a. For the *Label* use *Entity*
  - b. For the *Class Name* use *PGPImp-DemoPGP-Work-Entity*
  - c. Create and open
  - d. For Select class type use *Abstract*
  - e. For Created in version use the current version
  - f. For Parent class (directed) use *PegaPS-Work-Entity*
  - g. Save the rule
2. Use Create > SysAdmin > Class to create a new class
  - a. For the *Label* use *Entity Item*
  - b. For the *Class Name* use *PGPImp-DemoPGP-Work-Entity-Item*
  - c. Create and open
  - d. For *Select class type* use *Abstract*

- e. For Created in version use the current version
- f. For Parent class (directed) use *PGPImp-DemoPGP-Work-Entity*
- g. Save the rule

## Create Data Management Case Types

Use *Case types* -> *+Add a case type* to add new case types to the implementation layer. Use the Advanced Settings to select the directed and patterned ancestors for the new case type.

Choose the case type name carefully as it will be used to form the work class name for the case type. After the case type has been created, you may choose to change the *pyLabel* for the case type rule to something more specific.

The lists of ancestor classes to choose from may not show recently added abstract classes unless you have logged out since they were added or otherwise reset the cache.

### Create the Add Case Type

In PGP the *Add* case type for Entities collects data from the operator in a Page embedded on the work class. At an appropriate step in the case life cycle a PGP Persist component is placed which writes out the data from the embedded page on the work to a persistent data class using the persistence configuration from the *Database Table* rule for the embedded page's class.

1. Use *Case types* -> *+Add a case type* to create a new case type named *Add* (plan to change the label for the case type rule later)
  - a. Expand the Advanced section and specify the ancestry
  - b. Use *PGPImp-DemoPGP-Work-Entity-Item* as the directed ancestor
  - c. Use *PGPImp-DemoPGP-Work-Entity-Item* as the patterned ancestor
2. Create a Workflow for the case type
3. In the default *Create* stage
  - a. Use *+STEP* to add a Collect Information step under the *Create* stage
  - b. Name the step *Capture Item Details*

- c. From the step parameters panel (right) click *Configure the View*
  - 1. In the View Configuration add the *Item* field as a field group
  - 2. Select *ItemDetail* from the View dropdown to use it as the View for the *Item* field group
- 4. Use +STAGE to add a new stage named *Review*
  - a. Use +STEP to add a Collection Information step under the *Review* stage named *Review Item Details*
  - b. From the step parameters panel (right) click *Configure the View*
    - 1. From the step parameters panel (right) click *Configure the View*
    - 2. Select *ItemDetailRO* from the View dropdown to use it as the View for the *Item* field group
  - c. Use +STEP > ...More > Processes and select *Persist object*
  - d. From the step parameters panel (right) configure the *Persist object* component
    - 1. For *Type* use *Entity*
    - 2. For *Property* use ".item" (double quotes are significant)
- 5. Change the case type rule label for *PGPImp-DemoPGP-Work-Entity-Item-Add* from *Add to Add item*
- 6. Save the Case

## Create the Update Case Type

In PGP the Update case type for Entities solicits changes to the entity data from the operator and captures these changes in a Page or Page List embedded on the work class. At an appropriate step in the case life cycle a PGP Persist component is placed which writes out the data from the embedded page on the work to a persistent data class using the persistence configuration from the Database Table rule for the embedded page's class.

1. Use Case types > +Add a case type to create a new case type named *Update* (plan to change the label for the case type rule later)
  - a. Expand the Advanced section and specify the ancestry
    1. Use *PGPImp-DemoPGP-Work-Entity-Item* as the directed ancestor

2. Use *PGPImp-DemoPGP-Work-Entity-Item* as the patterned ancestor
2. Create a Workflow for the case type
3. In the default *Create* stage
  - a. Use +STEP to add a Collect Information step under the *Create* stage
  - b. Name the step *Edit Item Details*
  - c. From the step parameters panel (right) click *Configure the View*
    1. In the View Configuration add the *Item* field as a field group
    2. Select *ItemDetail* from the View dropdown to use it as the View for the *Item* field group
4. Use +STAGE to add a new stage named *Review*
  - a. Use +STEP to add a Collection Information step under the *Review* stage named *Review Item Details*
  - b. From the step parameters panel (right) click *Configure the View*
    1. In the View Configuration add the *Item* field as a field group
    2. Select *ItemDetailRO* from the View dropdown to use it as the View for the *Item* field group
  - c. Use +STEP > ...More > Processes and select *Persist object*
  - d. From the step parameters panel (right) configure the *Persist object* component
    1. For *Type* use *Entity*
    2. For *Property* use ".*Item*" (double quotes are significant)
5. Change the case type rule label for *PGPImp-DemoPGP-Work-Entity-Item-Update* from *Update* to *Update item*
6. Save the case
7. Edit the *pyDefault* data transform rule for the *Update* work type to set the initial values on the *.Item* page from lookup data page
  - a. Edit the *pyDefault* data transform rule for the *Update* work type to set the initial values on the *.Item* page from lookup data page
  - b. Add a *Set* action to the transform definition
    1. Use *.Item* as the *Target*
    2. Use *D\_Item[pyID:Param.EntityID]* as the *Source*
  - c. Save the data transform

## Create the View Case Type

In PGP the *View* case type for Entities shows a comprehensive snapshot of an entity instance's data.

1. Create a new Case Type named View
  - a. Expand the Advanced section and specify the ancestry
    1. Use *PGPImp-DemoPGP-Work-Entity-Item* as the directed ancestor
    2. Use *PGPImp-DemoPGP-Work-Entity-Item* as the patterned ancestor
2. Create a Workflow for the case type
3. In the Stage Name field for the Create stage use the smart-prompt to select *PegaPS-Work-Entity.View* from the list of available stage processes.
4. In the Stage Name field for the Create stage use the smart-prompt to select *PegaPS-Work-Entity.View* from the list of available stage processes.
5. Save the case
6. Override/SaveAs the PegaPS.Details section as *PGPImp-DemoPGP-Work-Entity-Item-View.Details*
  - a. Include the section *PGPImp-DemoPGP-Data-Entity-Item.Details* using the *.Item* property as the page context.
7. Save the section
8. Edit the *pyDefault* data transform rule for the *View* work type to set the initial values on the *.Item* page from lookup data page
  - a. Switch to the Parameters tab and add a parameter named *EntityID* of type String
  - b. Add a Set action to the transform definition
    1. Use *.Item* as the *Target*
    2. Use *D\_Item[pyID:Param.EntityID]* as the *Source*
9. Save the data transform

## Create the Search Case Type

In PGP the *Search* case type collects filter criteria from the user and retrieves a list of matching entities which may then be selected for further processing.

1. Create a new Case Type named Search
  - a. Expand the Advanced section and specify the ancestry
    1. Use *PGPImp-DemoPGP-Work-Entity-Item* as the directed ancestor
    2. ii. Use *PGPImp-DemoPGP-Work-Entity-Item* as the patterned ancestor
2. Create a Workflow for the new case type
3. In the Stage Name field for the Create stage use the smart-prompt to select the PegaPS-Work.Search stage from the list of available stages
4. Override/Save As the PegaPS-Work.Search section as *PGPImp-DemoPGP-Work-Entity-Item-Search.Search*
5. Edit the *PGPImp-DemoPGP-Work-Entity-Item-Search.Search* section
6. UseStructural > Embedded Section to include the section *PegaPS-Data.Search* in the dynamic layout
  - a. For *Page context* choose *Use clipboard page*
  - b. For *Class* enter *PGPImp-DemoPGP-Data-Entity-Item*
  - c. For *Clipboard page* select *.Item* from the list
  - d. For *Section* choose *By name* and select *Search* from the list
  - e. For *PARAMETER ShowSearchType* set *Value* to nothing
  - f. Click Submit
7. Save the section
8. Override/SaveAs the *PegaPS-Data.SearchCriteria* section as *PGPImp-DemoPGP-Data-Entity-Item.SearchCriteria*
  - a. Add the key properties for *PGPImp-DemoPGP-Data-Entity-Item* to this section
  - b. Save the section
9. Override/SaveAs the *PegaPS-Data.SearchResults* section as *PGPImp-DemoPGP-Data-Entity-Item.SearchResults*
10. Edit the *PGPImp-DemoPGP-Data-Entity-Item.SearchResults* section
11. Use *Structural -> Table* to add a table layout to the section and Configure the table
  - a. For *Source* choose *Data page*
  - b. For *Data page* select *D\_ItemList* from the list
  - c. For *Parameters* check *Pass the parameter page*
  - d. Submit
12. Save the section

## Enabling for App Studio

- Use views and design templates to construct the user interface
- Manage the new Entity's rules in Relevant Records so that appropriate choices will appear in the App Studio smart prompts.
- For appropriate case types set the *Show in 'New' menu* checkbox under Settings -> General to enable the case type to appear on +Create nav

## Appendix A - Rule Inventory

The following is a list of rules created in this exercise.

Applies -to	Rule Name	Rule Type	Description	Comment
	PGPImp-DemoPGP-Data-Entity	Class	Abstract	
	PGPImp-DemoPGP-Data-Entity-Item	Class	Concrete	Key EntityID
	PGPImp-DemoPGP-Data-Entity-Item	Database Table	Persistence	
PGPImp-DemoPGP-Data-Entity-Item	ItemName	Property	Text	
PGPImp-DemoPGP-Data-Entity-Item	ItemType	Property	Text	

Applies -to	Rule Name	Rule Type	Description	Comment
PGPImp-DemoPGP-Data-Entity-Item	ItemDetail	Section	View	
PGPImp-DemoPGP-Data-Entity-Item	ItemDetailRO	Section	View	
	D_Item	Data Page	Page	Source=Lookups
	D_ItemList	Data Page	Page List	Source=Report Definition
PGPImp-DemoPGP-Data-Entity-Item	ItemList	Report Definition		All Item Properties
PGPImp-DemoPGP-Work	.Item ( <i>PGPImp-DemoPGP-Data-Entity-Item</i> )	Page		
PGPImp-DemoPGP-Work	.ItemList ( <i>PGPImp-DemoPGP-Data-Entity-Item</i> )	Page		
	PGPImp-DemoPGP-Work-Entity	Class	Abstract	

<b>Applies -to</b>	<b>Rule Name</b>	<b>Rule Type</b>	<b>Description</b>	<b>Comment</b>
	PGPImp-DemoPGP-Work-Entity-Item	Class	Abstract	
	Add	Case Type		
	PGPImp-DemoPGP-Work-Entity-Item-Add	Class	Concrete	Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	pyDefault	Case Type		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	CreateForm_Default	Flow		Case Type
	D_Add	Data Page		Case Type
	D_AddList	Data Page		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	pyDefault	Data Transform		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	pySetFieldDefaults	Data Transform		Case Type

Applies -to	Rule Name	Rule Type	Description	Comment
PGPImp-DemoPGP-Work-Entity-Item-Add	ItemInformation_Flow	Flow		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	Create	Flow Action		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	Create	Section		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	ItemDetails	Flow Action		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	ItemDetails	Section		Case Type

## Implementing Pega Government Platform features

Implement and extend the features of Pega Government Platform to meet your business needs.

- **Review Checklist**
- **Consent Agreement**

- Document intake
- Verify document
- Nearby locations
- Evaluate
- Activity plan
- Assessments
- Multiple entity intake
- Case report
- Programs
- Operator Case Involvement (Touch) reporting

## Review Checklist

Review checklist is a reusable and generic component for a process through which an evaluator or manager can review the application progress based on various checklist items.

You can add a review checklist as a component from App Studio and you can configure the list for various use cases. Using the parameters of the component, you can choose a list to display in a case.

You can use review checklist in various intake applications of government business use cases, like grants application, issuance of certificates, license and permits, where a caseworker and manager need to complete different sets of checklist items.

IS Review checklist

**Checklist**

- Identification document
- Land registration document
- Finance document

[Cancel](#) [Save](#) [Submit](#)

## Using review checklist

Configure a checklist type and add the component to a step in a stage.

1. Configure a Checklist type that you can select to populate a Review Checklist. A Checklist type provides the checklist item rows that the checklist contains.
2. Add the review checklist process component, to a step, in a stage of your case type.

## Configuring a checklist type

Add a new checklist type with checklist items or update an existing checklist.

You can add a review checklist component in a case lifecycle as a step to introduce a process through which an evaluator or supervisor can review various to-do tasks that need to be completed as part of any application or case lifecycle.

1. In the navigation pane of App Studio, click PGP.
2. In the PGP explorer, click Review checklist.
3. Click the New checklist button to create a new review checklist.
4. In the Purpose box, enter a purpose for the checklist.
5. In the Checklist items section, click Add checklist item to add a new checklist item.
6. In the Label box, enter a label for the review checklist items. The corresponding code appears in the Code box.

7. Click Save.

## Configuring review checklist in case designer

Configure the review checklist component in App Studio to use in a case type, as a step, in any of the stages.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. In the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name for the stage in the text box.
6. Hover over Process, and then click STEP > More > Processes > Review Checklist.
7. Click the Select button.
8. In the Checklist details section, in the Source list, select the name of the checklist type that the Review Checklist shows:
  - To choose a specific checklist type, select By name, and then, in the Checklist type list, click one of the items.
  - To specify a property in the work class that supplies a specific checklist type at run time, select By property, and then enter a property value that is a valid checklist type.
9. In the Routing details area, select an operator:
  - To route the assignment to the current operator, click Current operator.
  - To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.

The screenshot shows the 'Step' dialog box. At the top, there's a 'Process name' field containing 'Review checklist'. Below it is a section titled 'Checklist details' which includes fields for 'Source' (set to 'By name') and 'Checklist type' (set to 'Grants'). Underneath this is a section titled 'Routing details' which includes a 'Route to' field with three options: 'Current operator' (radio button), 'Specific user' (radio button, selected), and 'Work queue' (radio button). There are also 'Source' and 'User' fields under 'Routing details'.

Figure: Section showing the routing details

10. Click Save.

## Extending review checklist

Implement a simple data-driven review component. If your application requires more robust question and answer capabilities, use the Pega Government Platform [Assessments](#) feature.

**Note:** The review checklist feature is provided as a simple reusable self-contained table-driven control. Use the review checklist feature as-is without rule or UI customization. Overriding the rules that compose review checklist

and *CheckListMap* is not recommended and may cause unexpected behavior during Pega Government Platform version upgrades.

## Consent Agreement

The Consent Agreement component displays an actionable consent agreement screen. The consent agreement screen contains the text stating the agreement clauses, a check box to agree, and a field to enter signature.

Government agencies can have various agreements, and terms and conditions that the constituents or the end users need to agree for various business use cases.

For example, for any contract award or for any grant application from the government, there must be an agreement between both the parties which is signed and agreed upon. The text and content must be different for different applications.

### New Grant

**Federal Consent Agreement**

Federal law requires this consent form to be signed and agreed by you. The Parties to this Consent Agreement are the Department and the Respondent, including Respondent's operating divisions and subsidiaries.

The Respondent agrees that no confidential information be used for personal use and benefits.

The Respondent certifies that as an individual, or any member of an entity, has not been convicted under Federal or State Law.

This Consent Agreement shall become binding on the Department only when the Assistant Secretary-Federal Law approves it by entering the Order, which will have the same force and effect as a decision.

I Agree



Clear

[Cancel](#) [Submit](#)

## Configuring consent agreement in case designer

Use the Consent Agreement process component in a case type as a step in any of the stages. Consent Agreement is a built-in dynamic PGP component that allows e-signatures and templates.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.

3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Consent agreement.
7. Click the Select button.
8. In the Consent agreement details section, in the Source list, select a source.
  - a. Select By name, and in the Consent agreement template list, select an agreement type.
  - b. Alternatively, select By property to specify a property on the work class that supplies the specific agreement type at run-time.
9. Select the Is e-signature required? check box to include an e-signature.
10. In the Routing details area, select an operator:
  - To route the assignment to the current operator, click Current operator.
  - To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.
11. Click Save.

**Result:**

Based on the business use case need, you can select the appropriate text template from the available templates and can include an e-signature.

## Extending consent agreement

Extend the Consent Agreement feature to add additional templates or edit the existing template text.



**Note:** To create new consent agreement templates, you must switch to Dev Studio.

1. In the navigation panel of Dev Studio, click Records.
2. Click Data Model > Property.
3. Click the filter for Property Name, and then search for ConsentAgreementTemplate.
4. Click the property.
5. In the Display and validation section, in the Prompt values area, create a new paragraph rule by clicking the Add a row icon.
6. In the Standard value box, enter a value, for example, CentralAgreementTemplate.
7. In the Prompt value box, enter or select the desired value.
8. Click Save.
9. In the navigation panel of Dev Studio, click Data types.
10. Click Option map.
11. On the Records tab, click Add record.
12. Enter the appropriate text for the ID, Code, Display value, Display order, and Field name / Group fields.

The screenshot shows a data management interface for an 'Option map'. The top navigation bar includes 'Data Type: Option map', 'PegaPS-Data-Config-OptionMap', 'PegaPS', 'Actions', 'Save', and a close button. Below the navigation is a menu bar with 'Data model', 'Usage', 'Sources', 'Records' (which is selected), 'Views', 'Test cases', and 'Settings'. The main area is titled 'Source' and shows a dropdown menu 'Local data storage' with an 'Actions' button. Below this is a toolbar with 'Export' and 'Import' buttons, a search bar, and a magnifying glass icon. A table lists three records:

ID	Code	Display value	Display order	Field name / Gr...	Disable object
ConsentAgreementTemplate_A	FederalConsentAgreement	Federal Consent Agreement 1		ConsentAgreementTemplate	
ConsentAgreementTemplate_B	StateConsentAgreement	State Consent Agreement 2		ConsentAgreementTemplate	
ConsentAgreementTemplate_C	CentralConsentAgreement	Central Consent Agreement 3		ConsentAgreementTemplate	

A blue link '+ Add record' is at the bottom left.

Figure: Adding a new consent agreement template

13. Click Save.

### Result:

After you refresh the *D\_PGPOptionMapList* and *D\_OptionMapList* data pages, the new consent agreement template is displayed in the Consent agreement template list.

## Document intake

Document intake enables you to attach the required documents for a particular category.

You can do the following from App Studio.

- Configure a document.
- Create a new document category or use an existing category.
- Add specific documents to a category.
- Configure other details such as, mandatory document, optional document, and so on.

For various applications, depending on the norms and policies, different set of documents need to be uploaded as a part of any case. To achieve different set of requirements for different applications, you can use this component and configure it accordingly.

New Grant

Passport	(i)	Optional	+ Attach
Driver's License	(i)	Optional	+ Attach
Bank Statements	(i)	Optional	+ Attach
Utility Bill	(i)	Optional	+ Attach

[Cancel](#) [Submit](#)

## Configuring Document intake in case designer

Use the Document intake process component in a case type as a step in any of the stages.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.

4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Document intake.
7. Click the Select button.
8. In the Document details section, in the Source list, select a source.
  - a. Select By name, and in the Purpose list, select a document from the list of required documents.
  - b. Alternatively, select By property to specify a property on the work class that supplies a specific document at run-time.
9. In the Routing details area, select an operator:
  - To route the assignment to the current operator, click Current operator.
  - To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.
10. Click Save.

#### Viewing the added documents in read-only view

11. Click the Views tab, and then click Create new view.
  - a. In the Search box, enter Intaken documents, and then click the Add to view icon.
  - b. In the Options list, select Read-only.
12. Click Submit.

### Editing or adding a new document purpose

Add a new document purpose with a list of documents.

For different applications, you may have to upload unique sets of documents. Using App Studio, you can edit an existing purpose and a list of documents, or add a new purpose and the corresponding list of documents.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click PGP, and then click Document intake.
4. Click a document category to edit it, or click the New document button to create a new document category.
5. In the Purpose box, enter the purpose of the document category.
6. In the Document items section, click Add documents to add them to the category.
7. Configure the document items as desired and click Save.

The screenshot shows the 'Document items' configuration for the 'Grants' purpose. It lists four document types: Passport, Driver's License, Bank Statements, and Utility Bill. Each item has a code, an optional checkbox for 'Is mandatory', and a description box with a trash icon.

Document type*	Code*	Is mandatory	Description
Passport	Passport	<input type="checkbox"/>	Upload Passport as a proof of identity
Driver's License	DriversLicense	<input type="checkbox"/>	Upload Driver's license as proof of age
Bank Statements	BankStatements	<input type="checkbox"/>	Upload the latest bank statement as financial proof document
Utility Bill	UtilityBill	<input type="checkbox"/>	Upload phone bill or electricity bill as address proof

[+ Add documents](#)

## Extending Document intake

Override the *UploadDocument (PegaPS-Work)* wrapper section that is shown to the user at run time to a specific application or to a specific case to update the UI as per your requirement.

## Verify document

The verify document component enables users to view the list of uploaded documents and take approval decision. You can configure a rejection action as a part of the document verification.

IS Verify document

Document review \*

Approve  Reject

Document type	Is mandatory	Attachment(s)
Passport	Optional	No attachments
Driver's License	Optional	No attachments
Bank Statements	Optional	No attachments
Utility Bill	Optional	No attachments

[Cancel](#) [Save](#) [Submit](#)

## Configuring Verify document in case designer

Configure routing details like the operator, stage, and so on based on approval decision. You can use the Verify document process component in a case type as a step in any of the stages.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Verify document.
7. Click the Select button.
8. In the Routing details area, select an operator:
  - To route the assignment to the current operator, click Current operator.
  - To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.
9. In the Reject stage list select a stage to configure the rejection stage when documents are rejected as a part of the review process.
10. Click Save.

## Extending Verify document

Add more document review options in addition to the existing options.

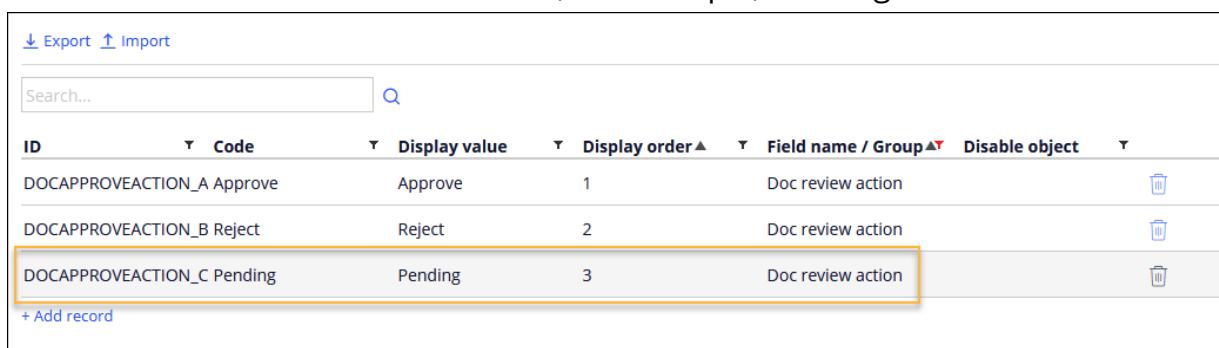
## VerifyDocument section (PegaPS-Work)

The wrapper section shown to the user at run time. You can override the section to a specific application or to a specific case to update the UI as per the requirement.

## VerifyDocument flow (PegaPS-Work)

The flow that holds the logic of routing to other stage when the verification is rejected. You can override the flow to a specific class or application as per the requirement.

1. In the navigation panel of Dev Studio, click Data types.
2. Click Data types > Option map.
3. Click the Records tab.
4. Click the filter for the Field name / Group column.
5. In the Search Text box, enter Doc review action and then click Apply.
6. Click Add record to add a new record, for example, Pending.



ID	Code	Display value	Display order	Field name / Group	Disable object
DOCAPPROVEACTION_A Approve		Approve	1	Doc review action	
DOCAPPROVEACTION_B Reject		Reject	2	Doc review action	
DOCAPPROVEACTION_C Pending		Pending	3	Doc review action	

7. Update the *VerifyDocument* flow to handle the newly added Pending action.
8. Click Save.

### Result:

After you refresh the *D\_PGPOptionMapList* and *D\_OptionMapList* data pages, the newly added record appears in the UI.

## Nearby locations

The Nearby locations process enables you to view important nearby places such as hospitals, schools, and so on that are supported by Google's Near Me service. You can also view the nearby cases and entities that are already in the application.

For example, if there is an application for school license, any reviewer can use this component to see if there are any bars or liquor stores in the nearby area and take the approval decision based on that. Also, for any investigation case for multiple events reported, the field investigators can use this component to check the event case in the nearby area.

**Nearby locations**

Nearby

Places

Find places close to this address [?](#)

1 Roger street, Boston, 02142

Search for  Display within radius  Range

Map data ©2021 Google Terms of Use

## Configuring Nearby locations

Use the Nearby locations process component in a case type as a step in any of the stages. You can configure the base location for searching nearby cases, entities, or places either with the current location, or using a property reference.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Nearby locations.
7. Click the Select button.
8. In the Location details section, in the Source list, select Current location to configure the current location as the base location for searching nearby cases, entities, or places.
9. Alternatively, select By property reference, and then enter or select a property from the combo box.
10. Select the Enable nearby result selection check box to select the nearby search results for further processing in the next steps or stages.
11. In the Routing details area, select an operator:
  - To route the assignment to the current operator, click Current operator.
  - To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.
12. Click Save.

13. To show a list of selected nearby places, click the Views tab, and then click Create new view.
  - a. In the Search box, enter Selected nearby places, and then click the Add to view icon.
14. To show a list of selected nearby cases or entities, Views tab, and then click Create new view.
  - a. In the Search box, enter Selected nearby cases and entities, and then click the Add to view icon.
15. Click Submit.

## Extending Nearby locations

Extend the Nearby locations component depending on the business use case and requirement.

### Enabling a new case type for Nearby cases search

Enable a new case type to retrieve on search.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Persist proximity location.
7. Click the Select button.
8. Expand Parameters and pass the property, which has the location to the process, as a parameter.



## Extending a new entity for nearby entity search

Add a new entity apart from the PGP-provided entities, as a part of nearby entity search, in the Nearby locations component.

For example, the *AddressList* property contains the addresses of a person. While saving the person details, the *PersistEmbeddedPage* activity (*CL:PegaPS-Data-Entity-Person*) runs which has the logic to save *AddressList* in *PegaPS-Data-Address* table.

Step	Label	Method	Step page	Description	Action
1.		Loop When >	Call LoadAddressListLatLong	Invoke activity to load GeoCodeAPI for address lat long	Jump
2.		Loop When >	Call SaveObject	.CommList	Save Communications
3.		Loop When >	Call SaveObject	.CommPreferenceList	Save comm preferences
4.		Loop When >	Call SaveObject	.EntityImage	Save image
5.		Loop When >		.SecurityQuestionList	Save security questions
1.		Loop When >	Property-Remove		Remove properties
2.		Loop When >	Call SaveObject		save question
6.		Loop When >	Call SaveObject	.LicenseList	Save Driving license
7.		Loop When >		.IdentifiersGroup	Save identifiers group list
1.		Loop When >	Property-Remove		Remove properties
2.		Loop When >	Call SaveObject		save identifier
8.		Loop When >	Call SaveObject	.CitizenshipList	Save Citizenship
9.		Loop When >	Connect-Wait		Wait for lat & long to load before persisting address
10.		Loop When >	Call SaveObject	.AddressList	Save Addresses

Show new entity as one of the type in nearby cases and entities

1. In the navigation panel of Dev Studio, click Data types.

2. Click Data types > Option map.
3. Click the Records tab.
4. Click Add record to add a new entity with Field Name / Group as Entity .

The screenshot shows a table with the following data:

ID	Code	Display value	Display order	Field name / Group	Disable object
Entity_A	Person	Person	1	Entity	
Entity_B	Business	Business	2	Entity	
Entity_C	Facility	Facility	3	Entity	
Entity_D	Vehicle	Vehicle	4	Entity	
Entity_E	Household	Household	5	Entity	

+ Add record

5. Save as *NearbyCasesList\_Ext* data transform and set *pyClassName* to new entity class when *pyLabel* is new entity.
6. Save as *FetchProximityEntity* report definition (*CL:PegaPS-Data-Entity-Facility*) into New entity class and remove the Name property from columns.
7. Create a new data page similar to *D\_FetchFacilityProximity* and call the report definition created in the previous step.
8. To show a different marker pin, on the map, for the new entity, save as *SetEntityMarker* data transform into the new entity class and set the *MarkerImage* property with any binary file.
9. Save as *SetNearbyAddressList\_Ext* activity, and load the data page created in step 7.
10. Save as *SetNearbyCasesEntitesResult\_Ext* data transform and append data page results to *NearbyCasesEntities* property.

## Evaluate

The Evaluate process performs the evaluation of a case type based on the evaluation criteria.

Evaluate provides an option to select evaluation type and routing type. Based on selected evaluation type, evaluation criteria are fetched. You can provide scores against

each criteria. The total scores are calculated based on its score and its weightage for each evaluation criteria.

**IS Evaluate case**

Evaluation criteria	Score	Maximum score	Weightage	Weighted Score
Financial criteria	<input type="range" value="48"/> 48	100	50	24
Personal criteria	<input type="range" value="72"/> 72	100	50	36

System recommendation  
**Approve**



60.00%

[Cancel](#)
[Save](#)
[Submit](#)

## Configuring Evaluate in case designer

Use the Evaluate process component in a case type as a step in any of the stages.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Evaluate.
7. Click the Select button.

8. In the Evaluation details section, in the Source list, select a source.
  - a. In the Source list, select By name.
  - b. In the Evaluation type list, select an evaluation type, for example, Grants.
  - c. Alternatively, in the Source list, select By property to specify a property on the work class that supplies the specific evaluation type at run-time.
9. In the Routing details area, select an operator:
  - To route the assignment to the current operator, click Current operator.
  - To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.
10. Click Save.

## Adding a new evaluation type

Add a new evaluation type with list of evaluation criteria.

1. In the navigation pane of App Studio, click PGP.
2. In the PGP explorer, click Evaluation criteria.
3. Click the New criteria button to create a new evaluation type. You can also update the existing evaluation criteria.
4. In the Evaluation type box, enter an evaluation type.
5. In the Evaluation items section, click Add evaluation item to add a new evaluation item.
6. Enter the Evaluation criteria, Evaluation code, Maximum score, and Weight.

 **Note:** The sum of all the weights of the criteria list must be equal to 100.

7. Click Save.

< All evaluations

### Grants

Evaluation type \* Grants

#### Evaluation items

Evaluation criteria*	Evaluation code*	Maximum score*	Weight*
Financial criteria	Financialcriteria	100.00	50.00 
Personal criteria	Personalcriteria	100.00	50.00 

+ Add evaluation item

## Extending Evaluate

Configure the system recommendation for the newly added type or for an existing type as you do it in the Grants example.

**Decision Table: Evaluation System recommendation [ Available]**

CL: PegaPS-Work ▾ ID: EvaluationSystemRecommendation RS: PegaPS:08-01-01

This record has 1 info warning (including 1 u

Table Results Parameters Pages & Classes Test cases Specifications History

Conditions Actions

Type	Percentage	>=	<=	Action
if	Grants	0	20	Reject
else if	Grants	21	40	Tentative
else if	Grants	41	100	Approve
otherwise				Approve

## Activity plan

Represent tasks and goals in an activity plan, and assign those tasks and goals to team members associated with the investigation. You can use activity plan as an optional action for an existing case as a case component. You can show the list of plans conducted on a case in a hierarchical table.

The activity plan landing page supports managing the plan templates, goals, and tasks. Managers or supervisors can create activity plan templates from the activity plan landing page. Each activity plan template can have any number of tasks and goals in it. You can develop activity plan templates using the *PegaPS-Work-Manage-ActivityPlan* case type. The captured plan template details are stored as instances of the *PegaPS-Data-Config-ActivityPlan* class.

Standalone tasks and goals can be created without creating the plan templates. Each goal should contain at least one task; you can create a custom task or add an existing task. When you create a task or a goal, PGP creates an instance of *PegaPS-Data-Config-Task* and *PegaPS-Data-Config-Goal* classes.

### New goal

Goal name \*

Description

Remaining: 300 characters

Due in \*      Unit \*

 Days

Task 1 Delete

Select existing task  Create custom task

Task \* ▼

+ Add task

## New Activity plan template

- X

### Activity plan

Activity plan  
Conduct investigation

Due in  
2 Days

Description  
Conduct investigation of the offence that is reported

### Plan tasks

Name	Description	Due in	
Conduct interview	Conduct interview of the involved entities	2 Days	

+ Add task

Goal 1: Capture evidence 2 tasks

Cancel Submit

New Activity plan template

[+ Add task](#)

Goal 1: Capture evidence

Goal Capture evidence	Description Capture evidence from the offence location	2 tasks
Due in 2 Days		

Tasks

Name	Description	Due in
Capture evidence from offence location	Capture evidence from offence location	2 Days
Capture evidence from CCTVs	Capture evidence from the near by CCTVs installed	2 Days

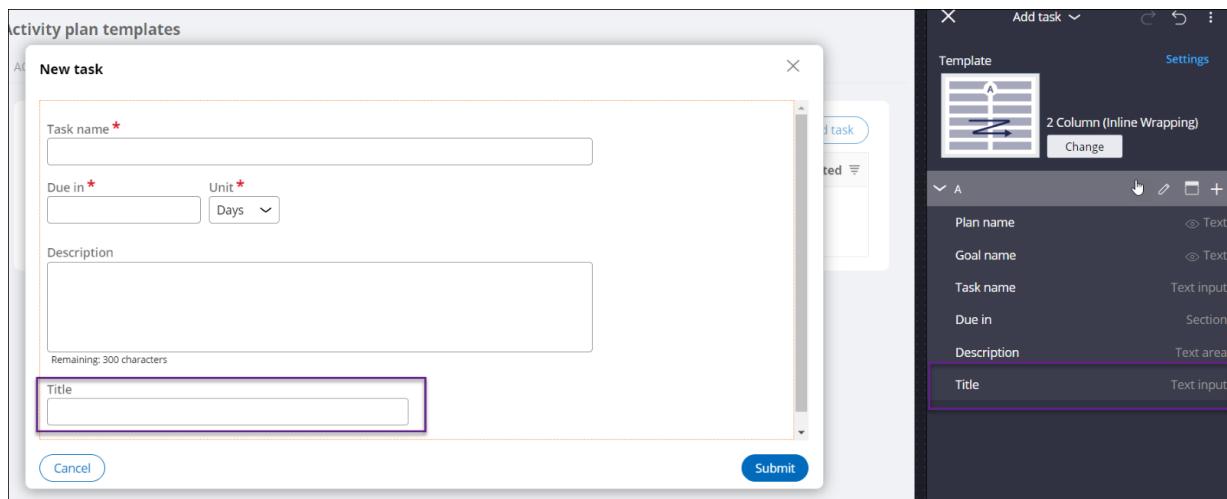
[+ Add task](#)

[Cancel](#) [Submit](#)

## Extending task details

Create, update, and delete tasks in activity plans. You can add custom tasks and create SLAs in days and hours.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the App Studio navigation pane, click Data, and then click the **Task** case type.
4. Click Add field to create a field, for example, Title.
5. In the Tasks tab of the Activity plan landing page, click the Add task button.
6. Navigate to the Add task view.
7. Click the Add icon and add a new field.

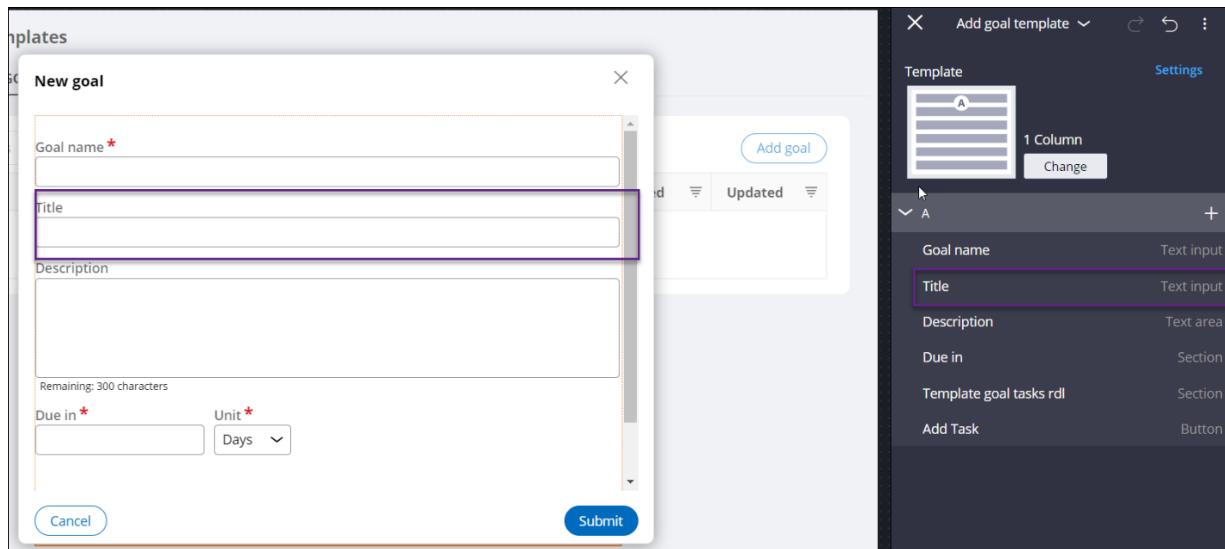


8. Click Save.

## Extending goal details

Create, update, and delete goals in activity plans. You can add custom tasks, add existing tasks, and create SLAs in days and hours for a goal.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the App Studio navigation pane, click Data, and then click the **Goal** case type.
4. Click Add field to create a field, for example, Title.
5. In the Goals tab of the Activity plan landing page, click the Add goal button.
6. Navigate to the Add goal template view.
7. Click the Add icon and add new field.

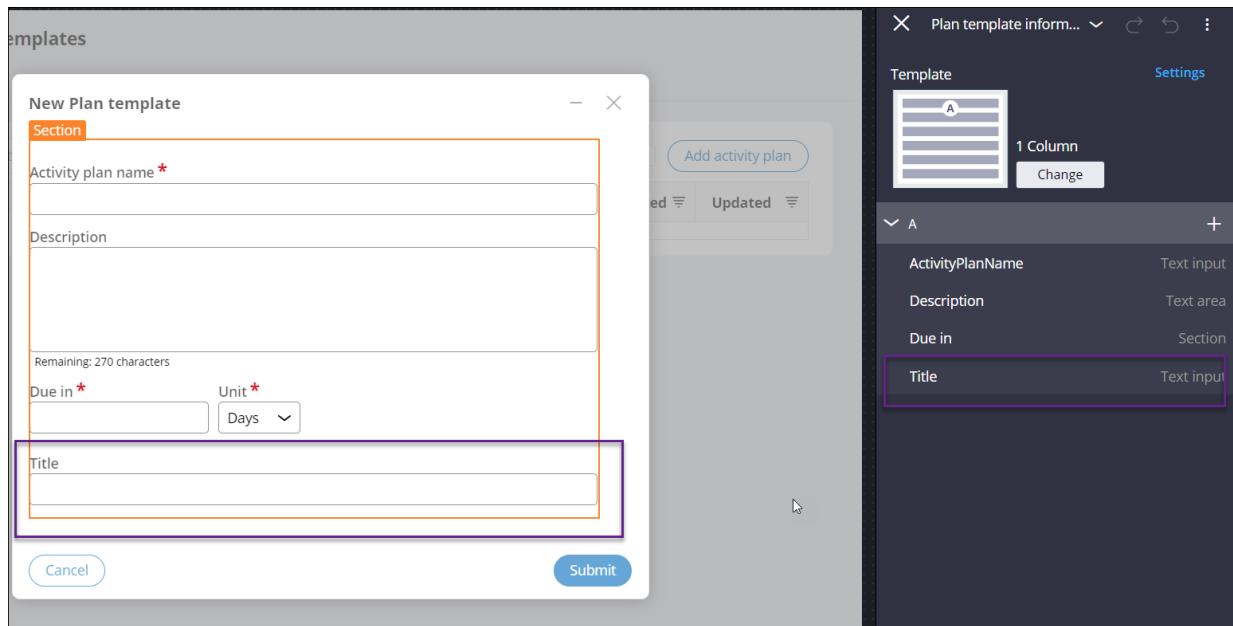


8. Click Save.

## Extending plan details

Add existing tasks or goals, create custom tasks or goals for the existing activity plans. You can segregate tasks for a goal, and goals for an activity plan so that you can add these plans to an investigation.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the App Studio navigation pane, click Data, and then click the **Plan** case type.
4. Click Add field to create a field, for example, Title.
5. Click the Add activity plan button from the landing page.
6. Navigate to the Plan template information view.
7. Click the Add icon and add a new field.



8. Click Save.

## Report for plans and option to convert them to template

Pega Government Platform provides a new tab called "Activity Plan Usage" in Activity Plan landing page. Under this tab, we can see the list plan templates that are used during creation of Activity plan case from Investigation or any other case. On expanding each row, the goals and tasks under each plan can be viewed. We can view different details about the templates like the name, description, status, how many times the template has been used, is it an existing or custom template, and if plan is a custom template giving an option to convert it to a new plan template in the system. Using search field, we can also search for any activity plan.

**Activity plan**

ACTIVITY PLAN TEMPLATES GOAL TEMPLATES TASK TEMPLATES ACTIVITY PLAN USAGE

Q Search activity plans 24 results

Plan	Description	Status	Usage count	Template	Action
> Test	Test description	Active	3	Existing	
> ch test task	ch test task	Active	2	Existing	
> Fishing	Fishing	Active	1	Existing	
> Activity plan fix	Activity plan fix		1	Custom	Convert to template
> Fishing	Fishing		1	Custom	Converted to template
> Custom Activity plan	Custom Activity plan		1	Custom	Convert to template
> Fishing	Fishing description		1	Custom	Convert to template
> Test plan custom 123	Test plan custom 123		1	Custom	Convert to template
> Sprint plan	Sprint plan		1	Custom	Convert to template
> Plan for investigation	Plan for investigation		1	Custom	Convert to template

1 2 3 Next

### Activity plan usage

## Include an existing template while converting to template

If implementation team has a new template field like SLADuration in their application. While converting custom plan to template, even these new template fields introduced in implementation layer should be populated and persisted in the manage template case. To support this procedure, follow these steps:

1. Create similar property (for example, SLADuration) in Index-PegaPS-Activity class.
2. Add the property for indexing in declare index rules – SaveIndexForActivityPlan, SaveIndexForGoal, and SaveIndexForTask.

**Declare Index: Save index for activity plan [Available]**

CL: PegaPS-Work-ActivityPlan ID: SaveIndexForActivityPlan RS: PGPCosmos:08-08-01

Source page context class  
PegaPS-Work-ActivityPlan

Index class to write  
Index-PegaPS-Activity

**Properties for indexing and mapping**

	Source class property	Mapping	Index class property
1	.pyLabel	Equals ▾	.pyLabel
2	.Description	Equals ▾	.Description
3	.TemplateSelection	Equals ▾	.SelectedTemplate
4	.Plan.pzInsKey	Equals ▾	.TemplateKey
5	.Plan.SLAType	Equals ▾	.SLAType
6	.Plan.SLAUnit	Equals ▾	.SLAUnit
7	.Plan.SLADuration	Equals ▾	.SLADuration
8	.pxObjClass	Equals ▾	.CaseObjClass
9	.pxCoverInsKey	Equals ▾	.CoverKey
10	.pxApplication	Equals ▾	.Application

*Declare index rule - SaveIndexForActivityPlan*

**Declare Index: Save index for goal [Available]**

CL: PegaPS-Work-Goal ID: SaveIndexForGoal RS: PGPCosmos:08-08-01

Source page context class  
PegaPS-Work-Goal

Index class to write  
Index-PegaPS-Activity

**Properties for indexing and mapping**

	Source class property	Mapping	Index class property
1	.pyLabel	Equals ▾	.pyLabel
2	.Description	Equals ▾	.Description
3	.TemplateSelection	Equals ▾	.SelectedTemplate
4	.Goal.SLAType	Equals ▾	.SLAType
5	.Goal.SLADuration	Equals ▾	.SLADuration
6	.Goal.SLAUnit	Equals ▾	.SLAUnit
7	.Goal.pzInsKey	Equals ▾	.TemplateKey
8	.pxObjClass	Equals ▾	.CaseObjClass
9	.pxCoverInsKey	Equals ▾	.CoverKey

*Declare index rule - SaveIndexForGoal*

Declare Index: Save index for task [ Available]

Source page context class  
PegaPS-Work-Task

Index class to write  
Index-PegaPS-Activity

**Properties for indexing and mapping**

Source class property	Mapping	Index class property
1 .pyLabel	Equals ▾	.pyLabel
2 .Description	Equals ▾	.Description
3 .TemplateSelection	Equals ▾	.SelectedTemplate
4 .Task.pzInsKey	Equals ▾	.TemplateKey
5 .Task.SLAType	Equals ▾	.SLAType
6 .Task.SLADuration	Equals ▾	.SLADuration
7 .Task.SLAUnit	Equals ▾	.SLAUnit
8 .pxObjClass	Equals ▾	.CaseObjClass
9 .pxCoverInsKey	Equals ▾	.CoverKey
10 .pxApplication	Equals ▾	.Application
11 .pxCreateOperator	Equals ▾	.pxCreateOperator

*Declare index rule - SaveIndexForTask*

3. Update the following report definitions to include the property as a column.
  - a. GetChildCasesListNew
  - b. GetCustomChildCases
  - c. GetKeysUsingTemplate
  - d. GetNewPlanDetails

Report definition: Get child cases list new [ Available]  
 CL: Index-PgaPS-Activity ID: GetChildCasesListNew RS: PGPCosmos:08-08-01  
 This record has 1 info warning (including 1 unjustified) [View](#)

Edit columns

Column source	Column name	Summarize	Sort type	Sort order
.pyLabel	Label	<blank>	<blank>	
.TemplateKey	Template key	<blank>	<blank>	
.SelectedTemplate	Selected template	<blank>	<blank>	
.Application	Application name	<blank>	<blank>	
.Description	Description entered	<blank>	<blank>	
.pxInsIndexedKey	Count	Count	<blank>	
.pxInsIndexedClass	Class of indexed instance	<blank>	<blank>	
.pxInsIndexedKey	Key of indexed instance	<blank>	<blank>	
<b>.SLADuration</b>	<b>Sla duration</b>	<b>&lt;blank&gt;</b>	<b>&lt;blank&gt;</b>	
.SLAType	SLA type	<blank>	<blank>	

*Report definition - GetChildCasesListNew*

#### 4. Add a new step in GetPlanDetails data transform and set the property.

Data Transform: Get plan details [ Available]

Step	Action	Value	Condition	Target	Properties
• 1.6	Set	.SelectedTemplate	equal to	.SelectedTemplate	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.7	Set	.TemplateKey	equal to	.TemplateKey	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.8	Set	.Count	equal to	.pySummaryCount(1)	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.9	Set	.pxInsIndexedKey	equal to	.pxInsIndexedKey	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.10	Set	.Status	equal to	.Status==""?pxPages(A):	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.11	Set	.pxInsIndexedClass	equal to	.pxInsIndexedClass	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.12	Set	.Prefix	equal to	""	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
<b>• 1.13</b>	<b>Set</b>	<b>.SLADuration</b>	<b>equal to</b>	<b>.SLADuration</b>	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.14	Set	.SLAType	equal to	.SLAType	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.15	Set	.SLAUnit	equal to	.SLAUnit	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.16	Set	.pxPages(Plan)	equal to	.pxPages(Plan)	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.17	Set	.pxIndexCount	equal to	.pxIndexCount	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>
• 1.18	Set	.pxIndexPurpose	equal to	.pxIndexPurpose	<a href="#">Select values +</a> <a href="#">Edit</a> <a href="#">Delete</a>

[+](#) [Collapse All](#) [Expand All](#)

*Data transform - GetPlanDetails*

#### 5. Add a new step in SetIndexDetailsToConfig data transform and set the property.

**Data Transform: Set index details to config [Available]**  
**CL:** PegaPS-Data-Config **ID:** SetIndexDetailsToConfig **RS:** PGPCosmos:08-08-01

This record has 1 info warning (including 1 unjustified) [View](#)

[Definition](#) [Parameters](#) [Pages & Classes](#) [Test cases](#) [Specifications](#) [History](#)

	Action	Target	Relation	Source
• 1	Comment	Setting index details to primary		
• 2	Set	.Label	equal to	IndexPage.pyLabel <a href="#">Select values +</a> <a href="#">⚙️</a>
• 3	Set	.SLADuration	equal to	IndexPage.SLADuration <a href="#">Select values +</a> <a href="#">⚙️</a>
• 4	Set	.SLAUnit	equal to	IndexPage.SLAUnit <a href="#">Select values +</a> <a href="#">⚙️</a>
• 5	Set	.Description	equal to	IndexPage.Description <a href="#">Select values +</a> <a href="#">⚙️</a>
• 6	Set	.SLAType	equal to	IndexPage.SLAType <a href="#">Select values +</a> <a href="#">⚙️</a>
• 7	Set	.InsKey	equal to	IndexPage.pxInsIndexedKey <a href="#">Select values +</a> <a href="#">⚙️</a>
• 8	Set	.pzInsKey	equal to	IndexPage.TemplateKey <a href="#">Select values +</a> <a href="#">⚙️</a>

[Collapse All](#) [Expand All](#)

Call superclass data transform [⚙️](#)

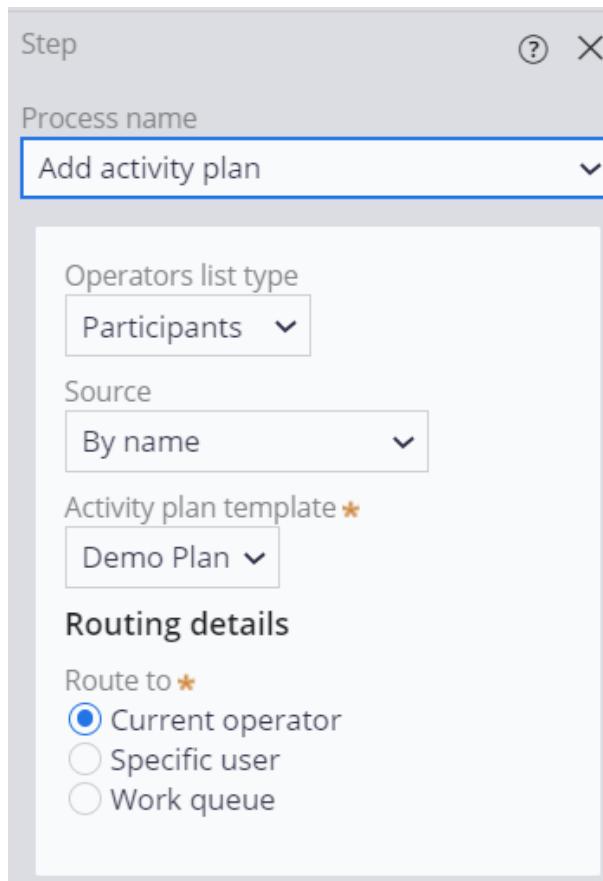
*Data transform - setIndexDetailsToConfig*

## Configuring activity plan in case designer

Create an activity plan component as a step in an investigation case. You can add from the existing activity plan template, create a custom activity plan, and delete or update activity plans for the investigation.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click Step More Processes Add activity plan.
7. Click the Select button.
8. In the add activity plan details section, in the Operators list type, select a type.

- a. When Participants is selected, the list of work parties of the case type from where the plan is launched are populated while assigning the goals and tasks.
  - b. Alternatively, select All operators to show all the available operators in the system while assigning the goals and tasks.
9. In the Source, Select a source.
- a. Select By Name, Select existing plans available in the system from the Activity plan template dropdown. This option is used to populate the plan details based on the selected plan in dropdown instead of selecting the plan in every run.
  - b. Select By property reference, Select the property from the autocomplete. This option is used to populate the plan details based on the selected property value. Here property value will refer to the plan name.
  - c. Select By configuration set. This option is used to populate the plan details based on the configuration set.
10. In the Routing details area, select an operator:
- a. To route the assignment to the current operator, click Current operator.
  - b. To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - c. To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.
11. Click Save.



## Assessments

The Assessments feature helps you create new assessments, questions, or question pages. Use the Assessments component to configure and conduct assessments.

This screenshot shows the 'Assessment management' interface. On the left is a vertical toolbar with icons for search, add, home, and refresh. The main area has a header 'Assessment management'. Below it is a table with columns: Name, Last Update, Last Updated By, and Configure. One row is visible: 'Risk assessment' (Last updated 08/30/21 by ICMCosmosSysadmin), with a 'Configure' button. At the top right of the table are buttons for Group, Fields, Density, and Refresh.

Name	Last Update	Last Updated By	Configure
Risk assessment	08/30/21	ICMCosmosSysadmin	Configure

In Pega Government Platform, access to Assessment landing page is given to a specific access group, for example, the manager access group. To show the Assessment landing page for another access group, update the setting from the Portals &pages tab.

Name	All personas	CaseWorker	End users	Manager	Unauth	Unauthenticated	User Services
<b>Portals</b>							
Default channel	User Portal	Select	User Portal	User Portal	User Portal	User Portal	User Portal
User Portal	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Pages</b>							
Activity plan	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Administration	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assessment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

For Investigative Case Management, Assessment is added only for the ICM Manager access group.

## Summary of features

Configuration of assessment includes providing an SLA name and interpretation with minimum or maximum score range. The following are the typical steps in working with assessments:

1. Configure the assessment for SLA and interpretations.
2. Conduct an assessment from a case, for example, from an investigation case.
3. View the conducted assessments from a case with interpretation score and status.
4. Calculate risk.

## Configuring assessments in case designer

Use the Assessment process component in a case type as a step in any of the stages. You can create assessments by creating a questionnaire case type from App Studio.

After you add the conduct assessment process as a step, you should select the assessment.

- If you select the assessment by name, then the same assessment initiates every time the process is run.
  - If you select the assessment by property reference, you need to send the class name of the assessment that must be initiated.
  - If you do not select an assessment in the process configuration, when the process is run, you can select an assessment from the available list.
1. Log in to Pega Government Platform as an administrator.
  2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
  3. In the navigation pane of App Studio, click Case types.
  4. From the Case types column, click a case type, or click the New button to create a new case type.
  5. In the Case life cycle section, click Stage, and then enter a name in the text box.
  6. Hover over process, and then click +STEP > More > Processes > Conduct assessment.
  7. Click the Select button.
  8. From configuration panel, select the source of the assessment.

## Configuring interpretations

Configure SLA and interpretations based on score of the assessment answers.

Configure interpretations for the assessment.

Based upon the interpretation results, business users can take further decisions or perform a set of actions.

For example, in a loan application, if the assessment case is configured as: if the score of all the responses is 0-5, and the interpretation is set as low risk. For any score above 5, it is set as high risk.

Based on the interpretation, the approver can understand and decide whether granting the loan amount is risky or not.

1. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
2. Click the Preview button, and then click Config Assessment.
3. Click Configure, and then click Add interpretation.
4. In the SLA Name list, select the number of days from 1 through 4.
5. In the Min score field, enter the minimum score.
6. In the Max score field, enter the maximum score.
7. In the Interpretation field, enter the interpretation based on the minimum and maximum scores. For example, if the Min score is 0 and the Max score is 3, then the interpretation is Good.

## Extending assessments

Extend the assessment feature using Dev Studio.

### Adding a field to an assessment configuration

Add a new data field or property and extend an existing assessment.

1. In the navigation pane of Dev Studio, click Records.
2. Expand the SysAdmin category, and then click Class.
3. Click the filter for Class Name.
4. In the Search Text field, enter *PegaPS-Data-Config-Assessment* and click Apply.
5. Click the rule to open it.
6. Create a property in the *PegaPS-Data-Config-Assessment* class.
7. Include the newly created property in the AssessmentConfiguration (*Rule-PegaQ-Questionnaire*) section.
8. Update the *D\_AssessmentConfigurationList* data page report definition to include the newly created property.

## Adding a service level agreement (SLA) to an assessment configuration

Add a new SLA type to the existing SLA configurations for any assessment.

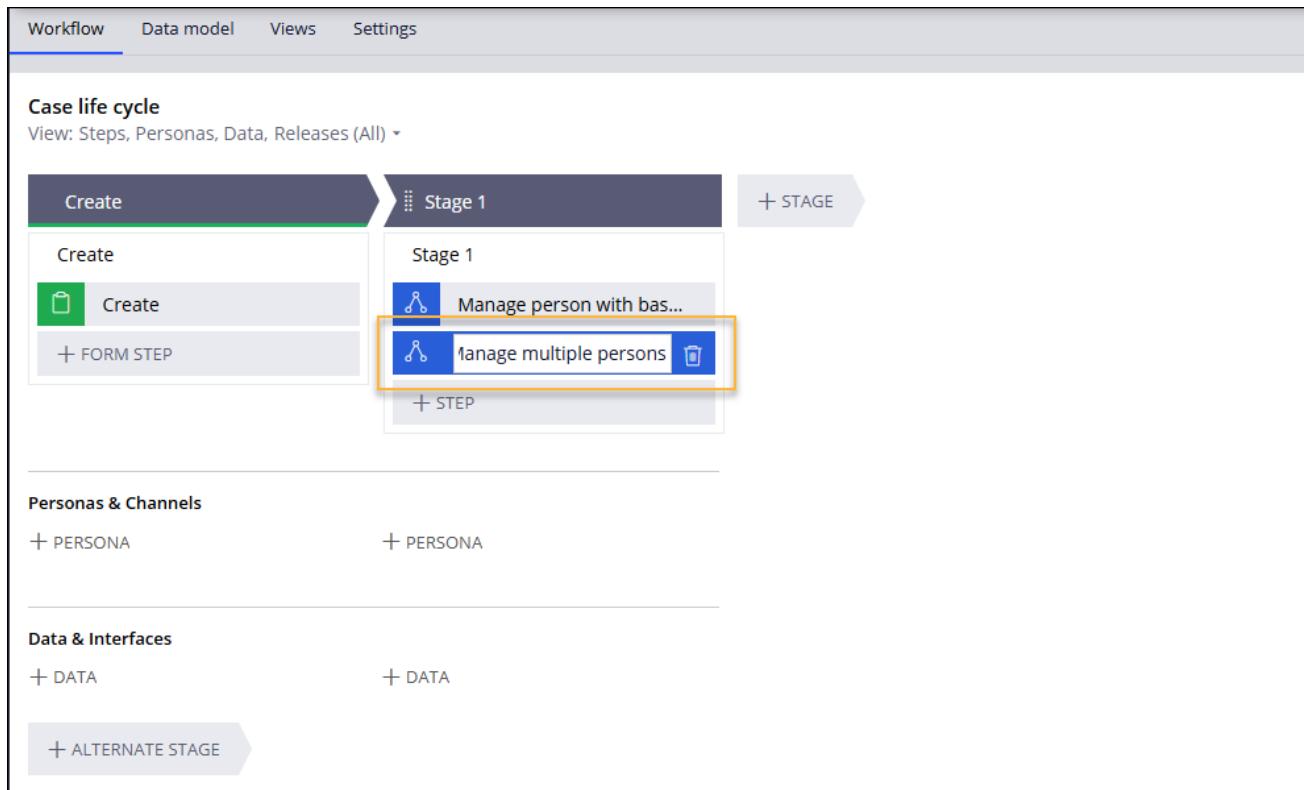
1. Create an SLA in the system that refers to OneDay in the *PegaPS-Work* class.
2. In the Explorer panel, click Data types > Option Map.
3. On the Records tab, add a row with the Field name/Group set to SLA.  
Ensure that the Code column value is the same as the ID of the newly created SLA.

## Multiple entity intake

The multiple entity intake feature enables you to capture information about multiple persons in a single step.

For many applications, there is always a need to capture information of all the family members as part of a single step. For each person, you can mention a relationship (for example, parent, child, grand parent, and so on).

While capturing the details of a person, you can either select an existing person in the system or create a new person.



## Multiple entity intake components

For multiple persons feature, Pega Government Platform offers components that you can use from App Studio for the following scenarios.

### Manage multiple persons (flow)

Use to capture the details of multiple persons: person ID, role, salutation, first name, last name, email, country, and address. This flow also validates the information added. After submission, the person details are persisted to the person table.

### Capture multiple persons details (flow action)

Use to capture details of multiple persons: person ID, role, salutation, first name, last name, email, country, and address.

### Persist multiple persons (flow)

Use to persist new persons details to person table.

## Multiple persons RO (section)

Use to display the list of persons captured.

## Implementing multiple entity intake

Add the multiple entity intake component when you build a case type. Many government applications require you to capture details of multiple persons as a part of the intake process, for example, multiple nominees, multiple beneficiaries, and so on.

## Configuring and extending multiple entity intake in a single screen

Configure and extend the multiple entity intake feature in a single screen.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Manage multiple persons.
7. Click the Select button.
8. In the Parameters section, in the Label box, enter a title for the component, for example, Dependent details.
9. In the Routing details area, select an operator:
  - To route the assignment to the current operator, click Current operator.
  - To specify a user or operator to whom you want to route this assignment, click Specific user, and then, in the User field, select a user or enter a property in the work class that supplies a specific operator ID at run time.
  - To route the assignment to a specific work queue, click Work queue, and then, in the Queue field, select a work queue name or enter a property in the work class that supplies a specific work queue at run time.

10. Click Save.

## Adding new fields to the intake screen

Update the *BasicPersonDetails (PegaPS-Data-Entity-Person)* report definition to add new fields to the intake screen for each person that is captured.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. From the Case types column, click Add person.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Manage person with basic details.
7. Click Save and run.
8. Click the Edit form icon to configure the view.
9. Click the section to which you want to add a field.
10. Click the Add icon.
11. Expand Fields, and then click the Add to view icon.
12. Click Submit.

## Updating relationship list values

Add new or update the existing values of the relationship list.

1. In the navigation panel of Dev Studio, click Data types.
2. Click Data types > Option map.
3. Click the Records tab.
4. Click the filter for the Field name / Group column.
5. In the Search Text box, enter PersonPerson and then click Apply.

A list of all the possible values for the relationships appears.

6. Click a record to open it, and update it as required.
7. If you want to add a new record, click Add record.
8. Click Save.

[Export](#) [Import](#)

ID	Code	Display value	Display order	Field name / Group	Disable object
PerToPer_A	Parent	Parent	1	PersonPerson	
PerToPer_B	Child	Child	2	PersonPerson	
PerToPer_C	Adopted Child	Adopted Child	3	PersonPerson	
PerToPer_D	Adoptive Parent	Adoptive Parent	4	PersonPerson	
PerToPer_E	Spouse	Spouse	5	PersonPerson	
PerToPer_F	Coworker	Coworker	6	PersonPerson	
PerToPer_G	Roommate	Roommate	7	PersonPerson	
PerToPer_H	Sibling	Sibling	8	PersonPerson	
PerToPer_I	Ex-Spouse	Ex-Spouse	9	PersonPerson	
PerToPer_J	Friend	Friend	10	PersonPerson	
PerToPer_K	Acquaintance	Acquaintance	11	PersonPerson	
PerToPer_L	Grand Parent	Grand Parent	12	PersonPerson	
PerToPer_M	Grand Child	Grand Child	13	PersonPerson	
PerToPer_N	Driver	Driver	14	PersonPerson	
PerToPer_O	Car Owner	Car Owner	15	PersonPerson	
PerToPer_P	Guardian	Guardian	16	PersonPerson	

[+ Add record](#)

## Configuring and extending multiple entity intake in separate screens

Configure and extend the multiple entity intake feature in separate screens.

The following are the supported components.

- Manage person with basic details
- Manage person with complete details

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.

4. From the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name in the text box.
6. Hover over process, and then click +STEP > More > Processes > Manage person with basic details.
7. Click the Select button.
8. Click Save.

Capturing the details of another person

9. Click the Data model tab, and then click the Add field button.
10. In the Field name field, enter the name for the field, for example, Father.
11. In the Type list, select Embedded data.
12. In the Data object list, select Person.
13. Click the Single record button, and then click Submit.

Add field to Add person X

Field name \*

Type

Data object \*

Options

Single record  
 List of records

> Advanced

Cancel Submit & add another Submit

14. Click the Workflow tab and configure the process that you added.

Process name  
Manage person with basic details

Entity ID  
 

Include in MyCases

Person page  
".Father"  


Label  
Father details 

**Routing details**

Route to \*

Current operator  
 Specific user  
 Work queue

## Entity ID

Enter or select the Entity ID of Father if it is available so that the component automatically gets the details from SOR and fills the appropriate fields. If this is left blank, a new entity is created with all the details submitted.

## Include in MyCases

Select this if you want to show the current case in MyCases of the entity.

## Person Page

Select the property of the Person type into which you must capture the details of the father and add quotes at both ends, for example, ".Father". The details are captured into the Father property.

## Label

Provide an instruction to the user, for example, Enter father details.

### Extending multiple entity intake in separate screens

To add new fields to the screens, update *BasicPersonInfo* or *PersonDetails* (*PegaPS-Data-Entity-Person*) based on the component used.

For more information on adding new fields to screens, refer to [Adding new fields to the intake screen](#).

## Case report

Case report is a reusable and generic component for a process. Using case report, a manager or reviewer can consolidate data from a case and can generate a Microsoft Word or PDF document.

You can add *create report* as a component from App Studio, and can generate a granular level report from case data. Pega Government Platform supports generating the details of case summary, interview, evidence, team members, assessments, and activity plans that are related to the case type. In addition to the above, the ICM application supports *subjects and external organizations* as well.

**New Grant**

**Create Report**

A report of case is an overview of all activity on a case. You will have the opportunity to create and edit your report before exporting for review.

Choose the document type to be generated **\***

PDF

Word

**Included sections**

Case summary

Interview

Evidence

Team

Assessments

Activity plans

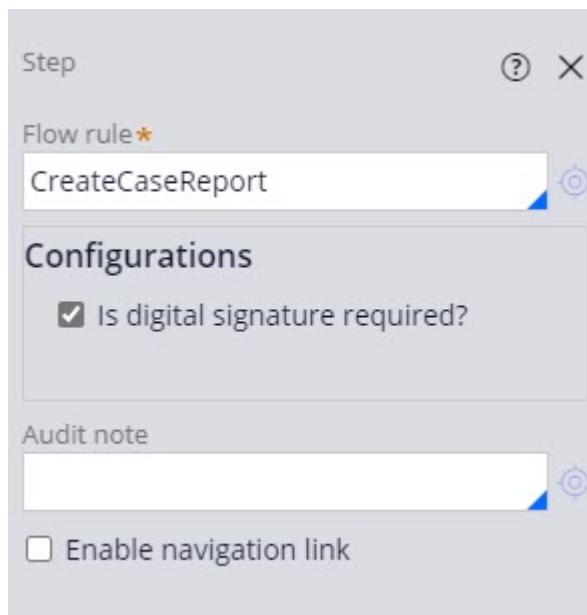
[Cancel](#) [Continue](#)

## Configuring case report in case designer

Configure the case report component in App Studio to use in a case type, as a step or an optional process, in a stage. You can use the case report component in case types that capture data related to interviews conducted, evidence, activity plans, and so on.

1. Log in to Pega Government Platform as an administrator.

2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. In the Case types column, click a case type, or click the New button to create a new case type.
5. In the Case life cycle section, click Stage, and then enter a name for the stage in the text box.
6. Hover over Process, and then click STEP > More > Processes > Create case report.
7. Click the Select button.
8. Under Configurations, click the checkbox "Is digital signature required?," to capture the digital signature.



*Configure case report*

9. Click Save to save the changes.

## Adding new data to the generated report

Create a case report with case details attached to the case type. The case report component is useful if the case type captures data related to interviews conducted, evidence, activity plans, and so on.

1. In the Dev Studio Explorer panel, click App.
2. Click the Classes tab, and in the Search box, enter PegaPS-Work.
3. Click Data Model > Data Transform > PreDocumentGeneration.



**Note:** If you are unable to view the **PreDocumentGeneration** data transform, click Show all results.

4. Click the data transform to open it.
5. Click the Save as button to save the data transform to the required class.
6. In the Data Transform Record Configuration section, in the Label box, enter a title.
7. In the Context section, select an application.
8. In the Apply to area, enter or select the required class.
9. In the Add to ruleset area, select a ruleset and the ruleset version.
10. In the Current work item section, enter or select a work item to associate.
11. Click the Create and open button.
12. Click the Add a row icon to add a new data transform to which you want to add data.
13. In the newly added data transform, in the Target box, enter or select the **.DocumentationReport** property.
14. In the Source box, append the required content in HTML format.
15. Click Save.

## Adding a new selection item to the report contents for list objects

Add a new case type component to the case report. During the document creation, the case type component and its details appear.

1. In the Dev Studio Explorer panel, click App.
2. Click the Classes tab, and in the Search box, enter PegaPS-Work.
3. Click User Interface > Section > Contents.
4. Click the Save as button.
5. In the Data Transform Record Configuration section, in the Label box, enter a title.
6. In the Context section, select an application.
7. In the Apply to area, enter or select a class.
8. In the Add to ruleset area, select a ruleset and the ruleset version.
9. In the Current work item section, enter or select a work item to associate.
10. Click the Create and open button.
11. Click the Add icon, and then click Pickers > Checkbox.
12. Double-click the newly added check box.
13. In the Cell Properties dialog box, enter the required details and then click Submit.
14. Update the *ShowItemsSelection (PegaPS-Work)* When condition with the new property created above.
15. Create a property in the *PegaPS-Work* class to hold the list of items for the user to select.
16. Update the *PreSelectItems (PegaPS-Work)* data transform to set the details to list property.
17. Update the *SelectItems (PegaPS-Work)* section to show the list of details for the user to select.
18. Follow the steps mentioned in the [Adding new data to the generated report](#) section, to include the selected items in the generated report.

## Using a Microsoft Word template

Create a new Word template while downloading the case report with details. The Word template helps in capturing key information and in maintaining consistency of the information.

1. Click Create > SysAdmin > WordTemplate.
2. In the Word Template Record Configuration section, in the Label box, enter a title.
3. In the Context section, select an application.
4. In the Apply to area, enter or select a class.
5. In the Add to ruleset area, select a ruleset and the ruleset version.
6. In the Current work item section, enter or select a work item to associate.
7. Click the Create and open button.
8. In the Template section, click Upload file to upload the required Word template, and then click Save.
9. In the Dev Studio Explorer panel, click App.
10. Click the Classes tab, and in the Search box, enter PegaPS-Work.
11. Click Process > Flow Action > Preview.
12. Click Save as to save the flow action into the case where you need a new template.
13. In the Flow Action Record Configuration section, in the Label box, enter a title.
14. In the Context section, select an application.
15. In the Apply to area, enter or select a class.
16. In the Add to ruleset area, select a ruleset and the ruleset version.
17. In the Current work item section, enter or select a work item to associate.
18. Click the Create and open button.
19. Click the Action tab.
20. In the Apply data transform box, enter or select PostPreview.
21. In the Post-processing section, in the Parameters area, enter the new template name and the class.
22. Click Save.

## Changing the PDF file styling

Change the styles in PDF document. The `<style>` tag defines the style information (CSS), where you mention how case details render in the PDF document.

When you generate a case report in the PDF file format, the *DocumentationReport* property which holds the complete markup stream of the PDF document, is passed as a parameter to the *GenerateCaseReportDocuments* activity through which the PDF document is generated.

The following is the styling CSS for PDF document in PGP and is set in step 4 of the *PostPreview (CL:PegaPS-Work)* data transform.

```
"<style> th{ align:justify; text-align: left; vertical-align: top; border-bottom: 1px solid black; } table{ border-collapse:collapse; } th,td{ padding:2px; font-size: 9pt; } td{ border-bottom: 1px dotted #000; font-weight:regular; text-align:top; vertical-align:top; } </style>"
```

1. Save the *PostPreview (CL:PegaPS-Work)* data transform to the implementation layer.
2. Update the new styling for the PDF document in step 4.

## Programs

As an agency, manage your licensing and certification processes by interacting with constituents and corporate entities. The program configurator omni-channel tool helps you meet customer needs more effectively and efficiently.

In program configurator, you can configure a new license by providing the required data for the license along with the image. The program configurator includes license configuration, a comprehensive application that streamlines licensing and certification application process from start to finish, including evaluation, document upload, and verification.

The program configurator enables you to do the following:

- Provide seamless, and simple licensing and certification processes for constituents, businesses, facilities, and vehicles.
- Reduce processing steps and minimize touch points to provide straight-through processing.
- Provide a modern government platform to meet future requirements and enable policy changes by business users.
- Reduce project implementation time and accelerate project delivery.
- Improve communication and interaction with constituents and businesses through omni-channel interactions.

## Program configuration

Government organization offers its citizens different services, different set of service details need to be created in the application. To achieve different set of details for different services, you can use this component and configure it accordingly.

The program configuration landing page can be accessed from app studio and supports configuring programs and licenses.

The screenshot shows the 'Program configuration' landing page. At the top, there is a navigation bar with a back arrow and the title 'Program configuration'. Below the title, there is a search bar with a dropdown menu. The dropdown menu has two options: 'Program' (which is selected) and 'License'. There is also a 'Category' dropdown and a 'Sub category' dropdown, both currently set to 'Select'. Below the search bar, there is a 'Search' button. The main area of the page displays four cards, each representing a program. Each card has a small icon at the top, followed by the name of the program: 'Program1', 'Program2', 'Program3', and 'Program4'. The cards are arranged in a grid format.

## Editing or adding a new program

For different applications, you may have various programs. Using App Studio, you can edit an existing program/license configuration details, or add a new program/license configuration with details.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click PGP, and in the quick launch menu, click Program configuration.
4. In the Category list, select Program or License.
5. Click any program or license to edit it, or click the New configuration button to create a new program or license.
6. Enter the details about the program and upload the image if required.
7. Click Save.

## Extending program configuration by adding new field

1. Create a new property in the *PegaPS-Data-Config-Program* class.
2. Update the *ProgramConfigurationDetails (CL:PegaPS-Data-Config-Program)* section and include the newly created property.

## **Adding a new program configuration and creating a new program case**

1. In the header of Dev Studio, click Data types.
2. Click Options > Add data type.
3. In the Label box, enter a name, for example, Grants.
4. Expand the Advanced section and in the Parent class box, enter or select the *PegaPS-Data-Config-Program* class, and then click Submit.

The screenshot shows the 'Add Data Type' dialog box. At the top, there are two radio buttons: 'New Data Type' (selected) and 'Existing Data Type'. Below this are fields for 'Label' (containing 'Grants') and 'Description' (containing 'Grants'). A 'Parent class' field contains 'PegaPS-Data-Config-Program'. The 'Identifier' field shows 'PegaPS-Data-Config-Program-Grants' with an 'Edit' link. Under 'Choose app layer', 'Impl PGP Cosmos' is selected. An 'Add to ruleset' dropdown is set to 'ImplPGPC'. At the bottom, there are 'Cancel' and 'Submit' buttons.

5. On the Records tab, click the Configure source button.
6. Select the Use as key check box for the pyID identifier.
7. In the Database box, enter or select **PegaDATA**.
8. Click Submit, and then click Close.
9. On the **Data model** tab, click the Add field button to add the new program configuration specific fields, for example, Land type.
10. Click Submit.
11. On the View tab, click the Program configuration details tile.

12. Click the Delete icon to delete any fields that are not applicable to the new program configuration.
13. Search for the fields that you have added in step 9, click the Add to view icon to add them to the view, and then click on Submit.

Field	Type	Options	
Name of the product	Text (single line)	Required	
Short name	Text (single line)	Required	
Description	Text (paragraph)	Required	
Sub category	Picklist	Required	
Application start date	Date & time	Required	
Application end date	Date & time	Required	
Duration type	Picklist	Required	
Duration units	Integer	Required	
Entity applicable	Picklist	Required	
Cost per item	Currency	Required	

14. Click Save to save the Grants data type.
15. In the header of Dev Studio, click Data types, and then click Option map.
16. On the Records tab, filter the Field name / Group column by typing ProgramCategory in the Search Text box.
17. Add the new program.

ID	Code	Display value	Display order	Field name / Group	Disable object
ProgramType_A	Program	Program	1	ProgramCategory	
ProgramType_B	License	License	2	ProgramCategory	
ProgramType_C	Grants	Grants	3	ProgramCategory	

18. Add new program sub categories. For subcategories, you must add the Field name column value that you added in step 17 as the Code column value.

ID	Code	Display value	Display order▲	Field name / Group ▾	Disable object
ProgramType_A	Program	Program	1	ProgramCategory	
ProgramType_B	License	License	2	ProgramCategory	
ProgramType_C	Grants	Grants	3	ProgramCategory	
GrantsCatergory_A	Industrial	Industrial	1	Grants	
GrantsCatergory_B	Commercial	Commercial	2	Grants	
<a href="#">+ Add record</a>					

19. Click Save to save the Option map data type.
20. Open the *D\_PGPOptionMapList* and *D\_OptionMapList* data pages and on the Load Management tab, click on Clear data page.
21. If you have deleted any required fields in step 12, delete the property in the *ValidateMandatoryDetails* validation rule.
22. In the header of Dev Studio, click Create > SysAdmin > Class.
23. Enter the details in the Class Record Configuration and the Context sections, and then click Create and open.

**Create Class**

**Class Record Configuration**

**Label\***  
Grants

A short description or title for this record

**Class Name**  
PegaPS-Data-Program-Grants

**Context**

Impl PGP Cosmos  
 Pega Government Platform Cosmos  
 Cosmos Rules  
 Pega Platform

Add to ruleset\*  
ImplPGPC

24. In the Created in version box, enter or select the ruleset version, on the History tab enter a description and then click Save.

**Edit Class: Grants**

ID: PegaPS-Data-Program-Grants RS: ImpIPG

General External Mapping Advanced History

Select class type  
Abstract ▾

**Settings**

Created in version  
01-01-01

**Class inheritance**

Find by name first (Pattern)  
Parent class (Directed)  
PegaPS-Data-Program

25. Open the Implementation Program work class and click Save as.
26. Enter the details and click Save.

Save As Class: Program

Class Record Configuration

Label\*

Grants

A short description or title for this record

Class Name

PGPGov-ImplPGPC-Work-Program-Grants

Context

Impl PGP Cosmos  
 Pega Government Platform Cosmos  
 Cosmos Rules  
 Pega Platform

Add to ruleset \*

ImplPGPC

27. Search and open the *SetProgramClasses (CL:PegaPS)* decision table, click Save as to save it to the implementation layer ruleset.
28. Add an else if condition for the new program configuration and in the Return column, enter the newly created Config class, for example, *PegaPS-Data-Config-Program-Grants*. Similarly, configure the new data and work classes.

Edit Decision Table: Set program classes [Available]				Delete	Add
CL:	PegaPS	ID:	SetProgramClasses	RS:	ImplPGPC:01-01-01
This record has 1 info warning (including 1 unjustified) <a href="#">Review/Edit</a>					
Table	Results	Parameters	Pages & Classes	Test cases	Specifications History
				Select values	Show conflicts
				Show completeness	Export
Conditions	Actions				
Class type	Category	Return			
if	Config	Program	D_AppContext.ClassNameValueGroup(DataProgramConfig)		
else if	Config	License	D_AppContext.ClassNameValueGroup(DataLicenseConfig)		
else if	Config	Grants	PegaPS-Data-Config-Program-Grants		
else if	Data	Program	D_AppContext.ClassNameValueGroup(DataProgram)		
else if	Data	License	D_AppContext.ClassNameValueGroup(DataProgramLicense)		
else if	Data	Grants	PegaPS-Data-Program-Grants		
else if	Work	Program	D_AppContext.ClassNameValueGroup(WorkProgram)		
else if	Work	License	D_AppContext.ClassNameValueGroup(WorkLicense)		
else if	Work	Grants	PGPGov-ImplPGPC-Work-Program-Grants		
otherwise			D_AppContext.ClassNameValueGroup(DataProgramConfig)		

## Operator Case Involvement (Touch) reporting

Review reports and analyze the day-to-day work of operators in the investigative case management applications. Touch reports provide insights about the work of an operator, on investigation cases, for a given time period.

- An operator touch report that shows the work for an operator for a given time period.
- A team touch report for a supervisor, to review the cases their team members worked on for a given time period. The team report access is restricted only to supervisors.
- A touch report for investigation cases, but can be extended to any case.

Touch points of operator for investigation		<a href="#">Edit report</a>	<a href="#">Actions</a>
Generated on November 17,2021 01:17:07			
Filtered by: Performer = ICM Cosmos Agent and Time = Current week and Work id Is not null and Object Class of this instance = PGPGov-ICMCosmo-Work-ICM-Investigation-Initiate			
Displaying 16 records			
Message		Time created ↓	
<b>INVESTIGATION-2006 : sad</b>			14
Notified the assignee Jason Lewis about the case ownership		11/16/2021 6:29 AM	
Transfer ownership task is performed.		11/16/2021 6:28 AM	
Transfer ownership task is performed.		11/16/2021 6:21 AM	
Request access R-129 from Jason is processed with action custom access.		11/16/2021 6:16 AM	
Created ROI.		11/16/2021 5:54 AM	
Created ROI.		11/16/2021 5:51 AM	
Assessment AS-2 has been conducted.		11/16/2021 5:50 AM	
Assessment AS-1 has been conducted.		11/16/2021 5:47 AM	
Child case sd P-2 has been resolved.		11/16/2021 5:46 AM	
Child case sd P-1 has been resolved.		11/16/2021 5:44 AM	
Conducted INTERVIEW-8.		11/16/2021 4:58 AM	
EVIDENCE-4 is added.		11/16/2021 4:56 AM	
SUBJECT-1011 is associated to INVESTIGATION-2006		11/16/2021 4:54 AM	
INVESTIGATION-2006 created.		11/16/2021 4:53 AM	
<b>INVESTIGATION-1001 : Theft at bank</b>			2
FACILITY-3 is associated to INVESTIGATION-1001		11/16/2021 6:32 AM	
Case has been accepted by ICM Cosmos Agent		11/15/2021 7:31 AM	

Touch points of team for investigation			
Generated on November 17, 2021 01:54:06			
Filtered by: Time = Current week and Work id Is not null and Performer Is not null and Object Class of this instance = PGPGov-ICMCosmo-Work-ICM-Investigation-Initiate			
Displaying 18 records			
ID <sup>1</sup> ↓	Name	Message	Time created <sup>2</sup> ↓
<b>ICM Cosmos Agent</b>			
INVESTIGATION-2006	sad	Notified the assignee Jason Lewis about the case ownership	11/16/2021 6:29 AM
INVESTIGATION-2006	sad	Transfer ownership task is performed.	11/16/2021 6:28 AM
INVESTIGATION-2006	sad	Transfer ownership task is performed.	11/16/2021 6:21 AM
INVESTIGATION-2006	sad	Request access R-129 from Jason is processed with action custom access.	11/16/2021 6:16 AM
INVESTIGATION-2006	sad	Created ROI.	11/16/2021 5:54 AM
INVESTIGATION-2006	sad	Created ROI.	11/16/2021 5:51 AM
INVESTIGATION-2006	sad	Assessment AS-2 has been conducted.	11/16/2021 5:50 AM
INVESTIGATION-2006	sad	Assessment AS-1 has been conducted.	11/16/2021 5:47 AM
INVESTIGATION-2006	sad	Child case sd P-2 has been resolved.	11/16/2021 5:46 AM
INVESTIGATION-2006	sad	Child case sd P-1 has been resolved.	11/16/2021 5:44 AM
INVESTIGATION-2006	sad	Conducted INTERVIEW-8.	11/16/2021 4:58 AM
INVESTIGATION-2006	sad	EVIDENCE-4 is added.	11/16/2021 4:56 AM
INVESTIGATION-2006	sad	SUBJECT-1011 is associated to INVESTIGATION-2006	11/16/2021 4:54 AM
INVESTIGATION-2006	sad	INVESTIGATION-2006 created.	11/16/2021 4:53 AM
INVESTIGATION-1001	Theft at bank	FACILITY-3 is associated to INVESTIGATION-1001	11/16/2021 6:32 AM
INVESTIGATION-1001	Theft at bank	Case has been accepted by ICM Cosmos Agent	11/15/2021 7:31 AM
<b>Jason Lewis</b>			
INVESTIGATION-2006	sad	Case owner has been changed to Jason Lewis	11/16/2021 6:30 AM
INVESTIGATION-2006	sad	Case ownership has been accepted by Jason Lewis	11/16/2021 6:30 AM

## Changing filter conditions

View the audit details of investigations for different time periods. By default, the work for the current week is displayed. You can change the time period.

By clicking Time filter and Select values, you get options to select different time periods or a particular date.

Touch points of operator for investigation

Generated

Filtered by: Performer = ICM Cosmos Agent and Time = Current week and Work id Is not null and Object Class of this instance = PGPGov-ICMCo

**Edit filter**

Time is equal  Current week

Filter Caption

Use null if empty

Ignore case

**Apply changes** **Cancel**

Displaying 16 records

Message

INVESTIGATION-2006 : sad

**Select values**

Time periods Calendar

**Calendar Year**

- Current Year
- Previous Year
- Previous 2 Years
- Next Year
- Current and Previous Year
- Current and Previous 2 Years
- Current and Next Year

**Calendar Quarter**

- Current Quarter
- Current and Next Quarter
- Current and Previous Quarter
- Next Quarter
- Previous Quarter

**Calendar Month**

- Previous Month

**Cancel** **Submit**

Time created ↓

By default, the report displays the audit details of all the investigations. To view the audit details of an investigation, click the Work ID filter and enter an Investigation ID.

Filtered by: Performer = ICM Cosmos Agent and Time = Current week and Work id Is not null and Object Class of this instance = PGPGov-ICMCo

**Edit filter**

Work ID is equal  "INVESTIGATION-2006"

Filter Caption

Use null if empty

Ignore case

**Apply changes** **Cancel**

Displaying 16 records

Message

By default, the report displays the work of all the team members. To display the work of a team member, click the Performerfilter and mention the name of the operator.

**Touch points of team for investigation**

Filtered by: Time = Current week and Work id Is not null and Performer = ICM Cosmos Agent and Object Class of this instance = PGPGov-ID

**Edit filter**

Performer is equal ICM Cosmos Agent

Filter Caption

Use null if empty

Ignore case

## Adding new column to reports

Display new columns for touch points of an operator for an investigation report. You can view other details of an investigation, such as reported activity, description, and so on.

1. Save the *InvestigationTouchReportForOperator* (CL: History-PegaPS-Work) report definition into the implementation ruleset.
2. Click Add column and enter the property that you want to display as a column and save.
3. Save the *InvestigationTouchReportForTeam* (CL: History-PegaPS-Work) report definition into the implementation ruleset.
4. Click Add column and enter the property that you want to display as a column and save.

## Creating similar reports for other case

As a manager or a team member, create similar reports for different case types.

1. Log in with an operator which points to the ICM implementation app.
2. In the user portal, in left navigation pane, click on Reports.
3. Click the All Reports tab, and search for the touch point of the operator for investigation and click that.
4. In the header, click Actions and then click Save as.
5. Enter the Title, Description and Category, and click on Submit.
6. Click the Object Class of this instance filter and provide the class value of the new case type for which similar report is required.
7. Click the Apply Changes button.
8. In the header, click the Done editing button. A new report for the new case gets created.

Similarly we can follow same steps for Touch points of team for investigation.

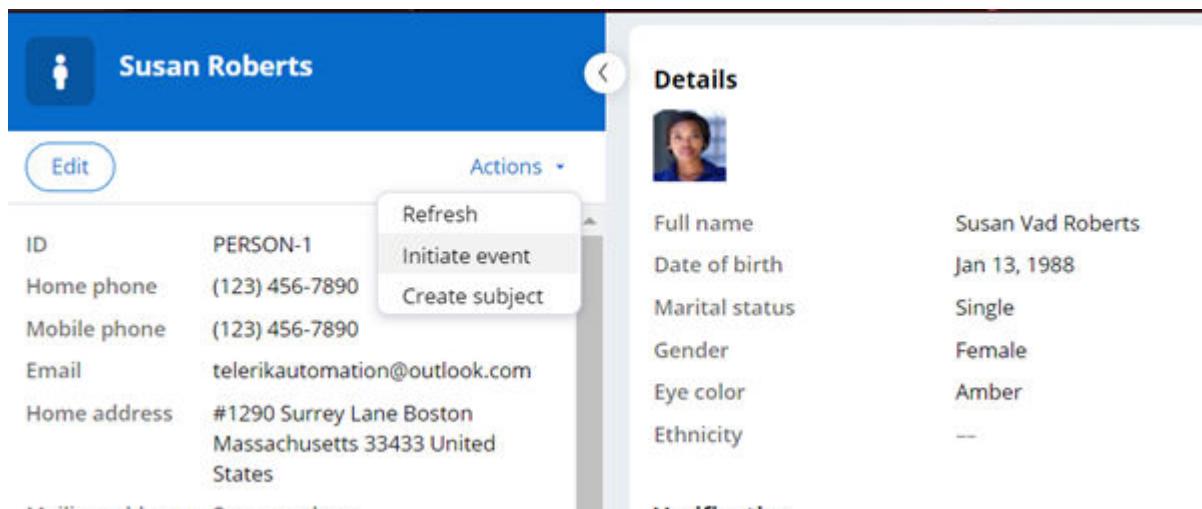
The screenshot shows the 'Touch points of operator for subject' report. The Data Explorer sidebar on the left lists various touch point categories like History class, Assignment Class, Flow Action, etc. The main report area displays a list of touch points, with a note about adding columns via drag-and-drop. The bottom section shows 5 records displayed, with sorting options for 'Message' and 'Time created'.

## Pega Government Platform case types

- **Initiate event**
- **Investigative Case Management**
- **Interview case**
- **Evidence case**
- **Application request**
- **Dashboard**
- **Search program Landing Page**
- **Manage associations**

## Initiate event

Intake specialists can initiate an event in the application, can perform and track actions. You can also initiate an event from the context of person entity and that event will get associate to the person.



The screenshot shows a person details screen for "Susan Roberts". The top navigation bar has a blue header with the name "Susan Roberts" and a person icon. Below the header is a toolbar with "Edit" and "Actions" buttons. The main area displays a table of personal information. To the right of the table is a "Details" panel showing more detailed information. A context menu is open over the "Actions" button, listing "Refresh", "Initiate event", and "Create subject".

ID	PERSON-1
Home phone	(123) 456-7890
Mobile phone	(123) 456-7890
Email	telerikautomation@outlook.com
Home address	#1290 Surrey Lane Boston Massachusetts 33433 United States

**Details**  
  

Full name	Susan Vad Roberts
Date of birth	Jan 13, 1988
Marital status	Single
Gender	Female
Eye color	Amber
Ethnicity	—

*Initiate event*

In order to use the event case in implementation layer, save as the D\_Event data page in the implementation layer and update the lookup with the implementation class name in the source.

## **Adding a field to the existing event details form**

When adding an event to the system, to add a field to be captured on the event details capture screen, complete the following steps.

1. Create a property in the *PegaPS-Data-Event* class.
2. Add the newly created property to the EventDetails (*PegaPS-Data-Event*) section.

## **Capturing an additional address for an event**

By default, when creating an event, one address is captured.

To capture an additional address, complete the following steps.

1. Create a record in the *AddressType* (*PegaPS-Data-Config-AddressType*) data type with the new address details.
2. Add a step in the *pySetFieldDefaults* (*PegaPS-Data-Event*) data transform for the new address type initialization.
3. Update the *EventInformation* (*PegaPS-Work*) section to include the new address type with the PageContext set as .Event.AddressList(2).

## **Adding a field to the existing review actions in the event review screen**

By default, the event review screen displays the following review actions in ICM application: open an investigation, Associate to an existing investigation, Need additional information, Issue warning, No Action-Invalid event, and No Action-Duplicate event. And in Pega government platform application, review screen displays the following review actions: Approve, Reject and Return.

To add a new review action for an event, complete the following steps.

1. Create a record in the Option Map (*PegaPS-Data-Config-OptionMap*) data type with the new review action.
2. Specify the field name as `C_EventReview` for the Pega Government Platform application.
3. Create a record in the ICM Option Map (*PegaPS-Data-Config-OptionMap-ICM*) data type with the new review action.
4. Specify the field name as `C_EventReviewAction` for the ICM application.
5. Open the *ReviewEvent* (*PegaPS-Work-Event-Initiate*) flow and click the Save as button.
6. Update the connectors that result from the decision shape with the newly introduced actions.

## View event

Pega Government Platform and Investigative Case Management users can view events and their associated data.

## Extending the Event view page

Extend the Event view page and display other properties of an event.

1. Define the new property in *PegaPS-Data-Event*, if the new property does not exist.
2. Add the created property in the *Details* (*PegaPS-Data-Event*) section.
3. To display a new utility, create the utility in the *PegaPS-Work-Event-Initiate* class, and add the utility to the *pyCaseMainInner* (*PegaPS-Work-Event-Initiate*) section.

## Search event

Use Search event to search the events in the system based on their basic information.

Search for events by using inputs such as Name, Event ID, Report type, State, and City.

## Adding new search criteria

Add a field to the search criteria for the Search event function.

1. Open the *SearchEventTable (PegaPS-Data-Portal)* section, and then add the required field to the section.
2. Add a parameter to the *D\_SearchEventList* data page and pass the value of the search text to this parameter in the above section.
3. To filter the records based on this field, add the parameter to the *SearchEventList* report definition, and add the parameter in the filter conditions.

## **Adding a field to the Search event search results**

To add a field to the search results for the Search event function, complete the following steps.

1. Add the required property to the columns space of the *EventDetails* report definition to retrieve it from the database.
2. Add the new column to the repeating grid in the *SearchResults (PegaPS-Data-Event)* section and refer to this new field.

## **Event creation through email**

The system can create an event work item for an inbound email from a reporter. The system can also parse the email and map the information from the email to an event work item.

## **Customizing the response message**

To customize the response message, complete the following steps.

1. Save as the following HTML rules:
  - EmailHeader (*Work-*)
  - EmailResponse (*Work-*)
2. Update the rules based on your requirements.

## Setting up an email listener for event creation

To configure an email listener in an implementation layer to initiate an event through email, complete the following steps.

1. Create an email account.
  - a. Click Records > Integration-Resources, then click Email Account.
  - b. Save as the *PegaPS-Work-ICM* email account in the implementation layer, for example, *PegaPS-ICMImpl-Work-ICM*.
  - c. Update the email address in the sender section and receiver section as necessary.
2. Save as the ICMIImplEventListener operator and change the access group to the administrator access group of the implementation application.
3. Configure the email listener.
  - a. Click Records > Integration-Resources, then click Email Listener.
  - b. Save as the ICMEventEmailListener listener to the implementation layer, for example, *ICMImplEventEmailListener*.
  - c. In the email account section of the listener, update it to use the email account that was created earlier, which in this example, is *PegaPS-ICMImpl-Work-ICM*.
  - d. Save as the service package (EmailApproval) and change the access group to the administrator access group of the implementation application.
  - e. Update the service package in the Listener properties section of the email listener.
  - f. Update the service class to the Event initiate class of the implementation application, for example, *PegaPS-ICMImpl-Work-Event-Initiate*.
  - g. Save as the service method CreatepyStartCase in the email listener to the implementation layer ruleset. Change the primary page class to the implementation layer's request access class, for example, *PegaPS-ICMImpl-Work-Event-Initiate*.
  - h. Update the requestor login section of email listener with the operator that you created earlier.
  - i. Save the email listener.

4. Add the email listener.
  - a. Open Admin Studio.
  - b. Click Resources, then click Listeners.
  - c. In the Start/restart listener menu, select the listener that was created in the previous step and click Apply.
  - d. Refresh the Active listeners section.

**Result:**

The listener displays in the list.

## Investigative Case Management

The Investigative Case Management application is meant for a different set of users, such as Investigators, ICMAgents, and so on. Use the application to manage the entire life cycle of an investigation.

The following list details some of the features of Investigative Case Management:

- Create investigations either standalone or from an event.
- Capture details for a case by using components such as interview, evidence, subject, and activity plans.
- Set the authorization of a user and how they search for, and view cases.
- Create case reports with features such as custom audits, visualizations, timeline control, and case closure.

### Investigation case

An Investigative Case Management user can create a standalone investigation case.

To use the Investigation case in the implementation layer, save as the D\_Investigation data page in the implementation layer and update the lookup with the implementation class name in the source.

## Adding a field to the existing investigation details form

When conducting an investigation, you must capture a certain number of details about the case.

Assume that you need to capture one more field when adding an investigation to the system. To add a field, complete the following steps:

1. Create the new property in the *PegaPS-Data-Investigation* class in the implementation layer.
2. Add the newly created property to the *CaptureInvestigationInformation* (*PegaPS-Data-ICM-Investigation*) section. Save as the section in the implementation layer.
3. Add a new property to an external organization:
  - a. Create the new property in *PegaPS-Data-Entity-Business*, if the property does not exist.
  - b. By adding a column in the table, add the newly created property in the *OrganizationList* (*PegaPS-Data-ICM-Investigation*) section.
  - c. Update the *BusinessListWithAddress* report definition with the new property created above.

## Viewing an investigation

As an Investigative Case Management user, view an investigation and the associated data collected during the investigation.

The case view shows the relevant information about the investigation in different tabs.

Investigation-specific data is displayed on the Details tab as an overview.

Add the property to display in the *Details* (*PegaPS-Data-ICM-Investigation*) section.

For information about adding a property, see [Adding a field to the existing investigation details form](#).

## Adding a search field

Search investigations in the system by using basic information such as investigation name, investigation ID, status, and reported activity.

To add a search field, complete the following steps.

1. Open the *D\_SearchInvestigationResults* data page and the *InvestigationResults (PegaPS-Work-ICM-Investigation-Initiate)* report definition. Add a parameter under the Parameters tab to both the rules.
2. On the Query tab of the *InvestigationResults (PegaPS-Work-ICM-Investigation-Initiate)* report definition, add the filter condition for the respective column.
3. Open the *SearchInvestigationTable* (*PegaPS-Data-Portal*) section. Open the table properties and pass the search text to the newly added parameter of the data page.
4. To show the new field in the results, add the new column to the table in the section with the property.

## Adding field in the search results

You can search the investigations in the system by using basic information such as date range, investigation ID, category, and so on.

To add a field in the search results, complete the following steps.

1. On the Query tab of the *InvestigationList* (*PegaPS-Work-ICM-Investigation-Initiate*) report definition, add the column to display in results.
2. Open the *SearchResults* (*PegaPS-Data-ICM-Investigation*) section. Add a column to the table and set it to the property to retrieve from the report definition.

## Setting up an email listener for investigation acceptance in an implementation layer

To configure an email listener in an implementation layer for the acceptance of investigation through email, complete the following steps.

1. Create an email account.
  - a. Click Records > Integration-Resources, and then click Email Account.
  - b. Save as the *PegaPS-Work-ICM-Investigation-Initiate* email account to the implementation class of the investigation, for example, *PegaPS-ICMImpl-Work-ICM-Investigation-Initiate*.
  - c. Update the email address as necessary in the sender section and the receiver section.
2. Save as the ICMInvestigationListener operator and change the access group to the administrator access group of the implementation application.
3. Configure an email listener.
  - a. Click Records > Integration-Resources, and then click Email Listener.
  - b. Save as the InvestigationAcceptEmailListener listener, for example, *InvestigationAcceptEmailImpl*.
  - c. In the email account section of the listener, update it to the email account that was created in step 1, for example, *PegaPS-ICMImpl-Work-ICM-Investigation-Initiate*.
  - d. Save as the service package (*InvestigationEmailDefault*) and change the access group to the administrator access group of the implementation application.
  - e. Update the service package in the Listener properties section of the email listener.
  - f. Update the service class to the Request access class of the implementation application, for example, *PegaPS-ICMImpl-Work-ICM-Investigation-Initiate*.

## Subject case

An investigator can add a new subject (which can be an entity, such as a person, business, vehicle, or facility) by creating a subject directly from the investigation or by creating a standalone subject and later associating it with an investigation. You can also create a new subject from the context of entities (Such as Person, Business, Facility and Vehicle) and that entity will get associated to the subject.

ID	PERSON-1
Home phone	(123) 456-7890
Mobile phone	(123) 456-7890
Email	telerikautomation@outlook.com
Home address	#1290 Surrey Lane Boston Massachusetts 33433 United States

**Details**

Full name	Susan Vad Roberts
Date of birth	Jan 13, 1988
Marital status	Single
Gender	Female
Eye color	Amber
Ethnicity	—

*Create subject*

To use the Subject case in the implementation layer, save as the D\_Subject data page in the implementation layer and update the lookup with the implementation class name in the source.

## Adding a field to the existing subject details form

To capture more details when adding a subject to the system, complete the following steps.

Add a new subject type.

- a. Add the new entity type as a record in the option map and set the Field name/ Group value as EntityType.
- b. Save as the SetICMEntityClass data transform in the implementation layer.
- c. Create a when rule, for example, IsPerson, and add another step for the new entity.
- d. Update the .ICMEntity page to update the class with the new entity class name.
- e. Save as the *ICMEntityDetails (PegaPS-Data-ICM-Entity)* section to the entity class and update it with the properties that need to be captured when creating the subject.
- f. Open the *NewSubject (PegaPS-Data-Entity)* section and click the Save as button.

- g. Save to the entity class and update it with the properties that you need to be capture when creating a new subject.
- h. Update the Identification (*PegaPS-Work-ICM-Subject-Initiate*) section by adding the *SearchKnownSubject* section with the new entity page context.
- i. Open the *SearchEntityCriteria* and *SelectKnownSubject* (*PegaPS-Data-Entity*) sections and click the Save as button.
- j. For implementing the select existing feature, save the sections to the entity class.

## View a subject

An Investigative Case Management user can view a list of subjects in the investigation case under Subjects.

An Investigative Case Management user can view a list of subjects in the investigation case under Subjects.

- To open the subject view page, click the Name link.
- To view the newly created entity details, create the *SubjectDetailsRO* view in the newly created entity class.

For more details, refer to *EntityDetails* (*PegaPS-Work-ICM-Subject-Initiate*) class.

## Adding a search field for a subject

You can search for a subject in the system by using basic information such as Subject ID, Type, Role, and so on.

To add a search field, complete the following steps.

1. Open the *D\_SearchSubjectResults* data page and the *SearchSubjectResults* (*PegaPS-Work-ICM-Subject-Initiate*) report definition. Add a parameter under the Parameters tab to both the rules.
2. On the Query tab of the *SearchSubjectResults* (*PegaPS-Work-ICM-Subject-Initiate*) report definition, add the filter condition for the respective column.

3. Open the *SearchSubjectTable (PegaPS-Data-Portal)* section. Open the table properties and pass the search text to the newly added parameter of the data page. If the new parameter does not display, reenter the data page name.
4. Add the new column to the table in the section with the property to show the new filed in the results.

## Adding a field in the search results for a subject

To add a field in the search results for a subject, complete the following steps.

1. On the Query tab of the *SearchSubjectList (PegaPS-Work-ICM-Subject-Initiate)* report definition, add the column to display in results.
2. Open the *SearchResults (PegaPS-Data-ICM-Entity)* section. Add a column to the table and set it to the property to retrieve from the report definition.

## Interview case

An Investigative Case Management user can conduct an interview for an investigation case or for a subject case by creating an interview directly from the investigation or subject.

Edit an interview by using the Edit interview feature, which you can use to add another participant, update the values that are captured, and so on.

## Adding a field to an existing interview details form

When conducting an interview for an investigation case or for a subject, you must capture a certain number of details about the case or subject.

Assume that you need to capture one more field when adding an interview to the system. To add a field, complete the following steps:

1. Create the property in the *PegaPS-Data-Interview* class.
2. Add the newly created property to the *InterviewInfo (PegaPS-Data-Interview)* section.

3. Add a property to interview participants:
  - a. Create the property in the *PegaPS-Data-Association* class.
  - b. Add the newly created property to the *ParticipantsList (PegaPS-Data-Association)* section.
4. Add an interview template:
  - a. Switch to App Studio and click the PGP wrench icon.
  - b. Click Interview templates and add new template to the existing list.

## Viewing an interview

An Investigative Case Management user can view an interview and the associated data that was collected during the interview.

1. Add the property to display in the InterviewInfoRO (*PegaPS-Data-Interview*) section.  
For information about adding a property, see [Adding a field to the existing interview details form](#).
2. Update the InterviewList (*PegaPS-Data-Interview*) report definition to retrieve the property.

## Extending the Interview case

The Interview case type can be extended to any other cases by completing the following steps.

1. Click Case types.
2. Click the vertical ellipsis, and then click Open.
3. Add the Interview case as a child case type.
4. Save your changes.
5. To show the list of interviews of a case type, include the *InterviewTab (PegaPs-Work)* section in the *pyCaseMainInner* section as a new tab.

## Evidence case

As an Investigative Case Management user, add evidence for an investigation case by creating an evidence case directly from the investigation.

## Extending the Evidence case

The Evidence case type can be extended to any other cases by completing the following steps.

1. Click Case types.
2. Click the vertical ellipsis and then select Open.
3. Add the Evidence case as a child case type.
4. Save your changes.
5. To show the list of evidences of a case type, include the *EvidenceTab (PegaPs-Work)* section in the *pyCaseMainInner* section as a new tab.

## Capture item from Evidence case

An Investigation Case Management user can capture the item details from the evidence case.

In evidence, only basic details of the item are captured using the Manage basic item component. In this component, item details are captured in a single screen. If you need to capture the full item details in a screen flow, then Manage Item component can be used.

To capture the item from evidence, Select the Capture Item check box in the first screen of evidence case and then item details will be captured in the next screen.

The details of the item captured will be displayed in the separate tab called Item in the evidence case. Also in the investigation case, under evidence tab, you can see the item in the separate column.

## Configuring capture item details case designer

Configure the item basic details component in App Studio to use in a case type, as a step or an optional process, in a stage. You can use the item basic details component in case types that capture basic data related to the item.

1. Log in to Pega Government Platform as an administrator.
2. In the header of your workspace, click the Switch Studio menu, and then click App Studio.
3. In the navigation pane of App Studio, click Case types.
4. In the Case types column, click evidence case type.
5. In the Case life cycle section, click Stage and enter a name for the stage in the text box.
6. Hover over Process, and then click Step > More > Process > Manage basic item.
7. Click Select.

## Extending Evidence data

To extend Evidence data, complete the following steps.

1. In the Dev Studio Explorer panel, click App. In the search field, enter PegaPS-Data-Evidence.
2. Right-click the class and create a property for the information that you want to add, for example, Complaints.
3. Include the new property in the section.
  - a. In the Application Explorer, find the *PegaPS-Data-Evidence* class.
  - b. Click User Interface > Section > EvidenceInfo.
  - c. Add the new property in the section.
4. Include the new property in the EvidenceRO section of the *PegaPS-Data-Evidence* class by repeating step 3.

## Viewing evidence

An Investigative Case Management user can view evidence and the associated data that was collected for the evidence.

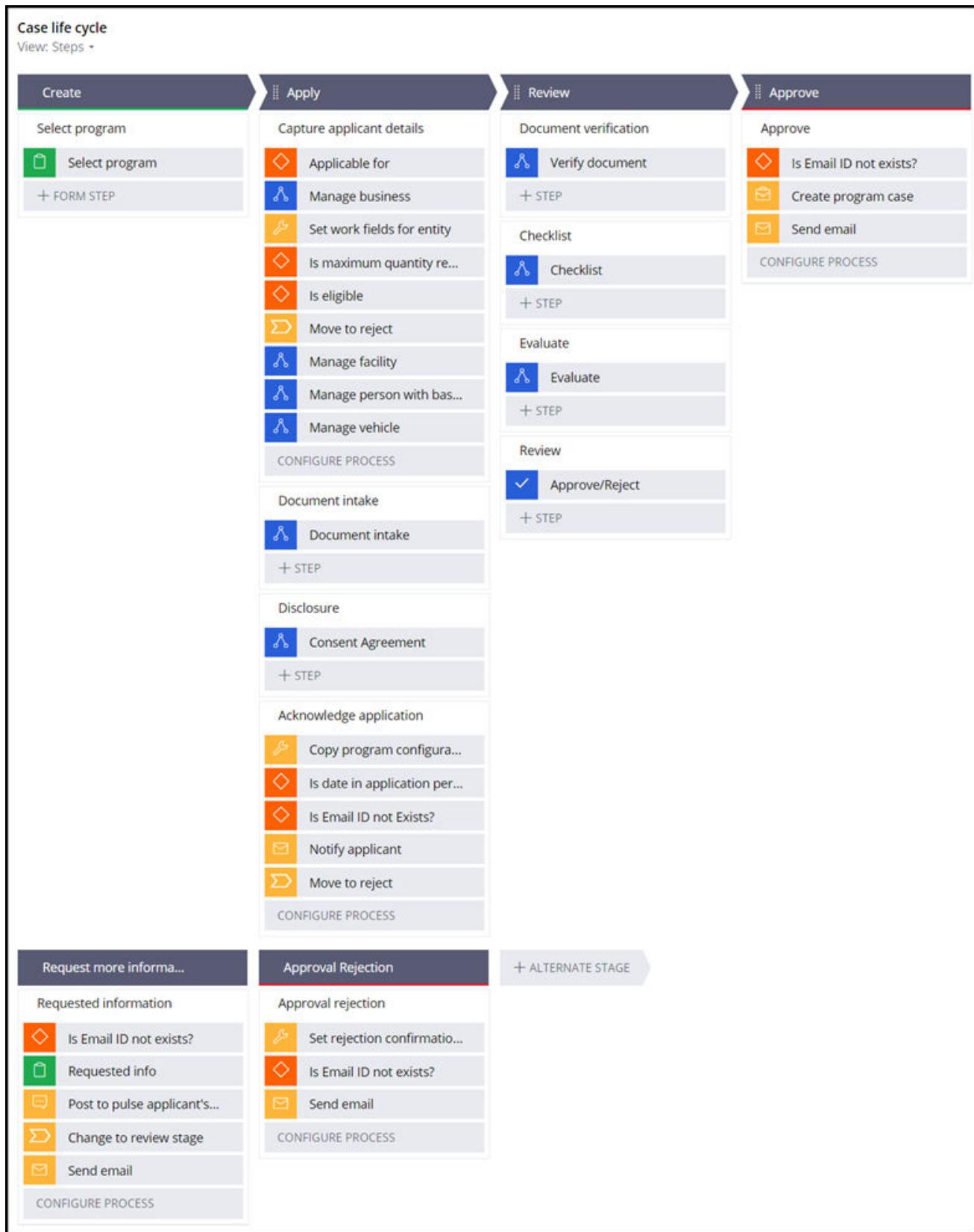
1. Add the property to display in the EvidenceInfoRO (*PegaPS-Data-Evidence*) section.
2. Update the EvidenceList (*PegaPS-Data-Evidence*) report definition to retrieve the property.

## Application request

The application request case type enables you to apply for a program or any of its descendants, for example, a license. Using application request, you can bring all the services of your organization under one case type.

The application request case type allows business users to streamline application processes, such as capturing entity details, document intake, consent agreement, and so on from start to finish. The business managers can review and approve the attached or uploaded documents, review the checklist, evaluate the process, and approve the application.

Users can apply for a license/program, which is applicable for all entities except household, through Application request. After a user applies for a license/program it is sent for review.

*Application request*

If the application is approved, an instance of license/program is created and these are listed in “Programs” tab of entity.

The following image shows programs/licenses of a person:

ID	Name	Start date	End date	Status
P-2001	Food program	Dec 26, 2021	Jan 3, 2022	Active
L-2001	Driving license	Mar 31, 2022	Mar 31, 2023	Active

*Programs tab*

## Dashboard

The dashboard in the portals provides users with a diverse set of reports based on application and their access group.

- **Reports on Investigation**
- **Reports on Event**
- **Reports on Application request**
- **Nearby cases chart**
- **Add a new report/chart to dashboard**

## Reports on Investigation

Users logging with ICMCosmosSupervisor,ICMCosmosAgent access have reports related to investigations in dashboard.

The following are the reports:

- Investigation by status: The pie chart represents the number of investigations for a status.
- To make changes, open “InvestigationByStatus”(*CL:PegaPS-Work-ICM-Investigation*) section and “InvestigationByStatus” (*CL: PegaPS-Work-ICM-Investigation*) report definition.

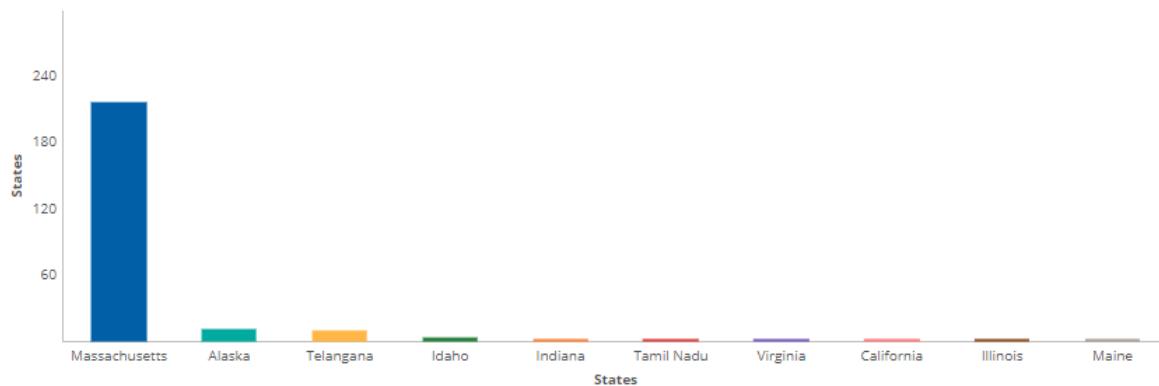
## Reports on Event

Users with access other than “ICMCosmosSupervisor,” “ICMCosmosAgent” have reports related to events in dashboard.

The following are the reports:

- Events by top 100 locations: The following is a column chart representing top 10 states which have most events with all statuses except “resolved-Cancelled”. Events with no address information are excluded.

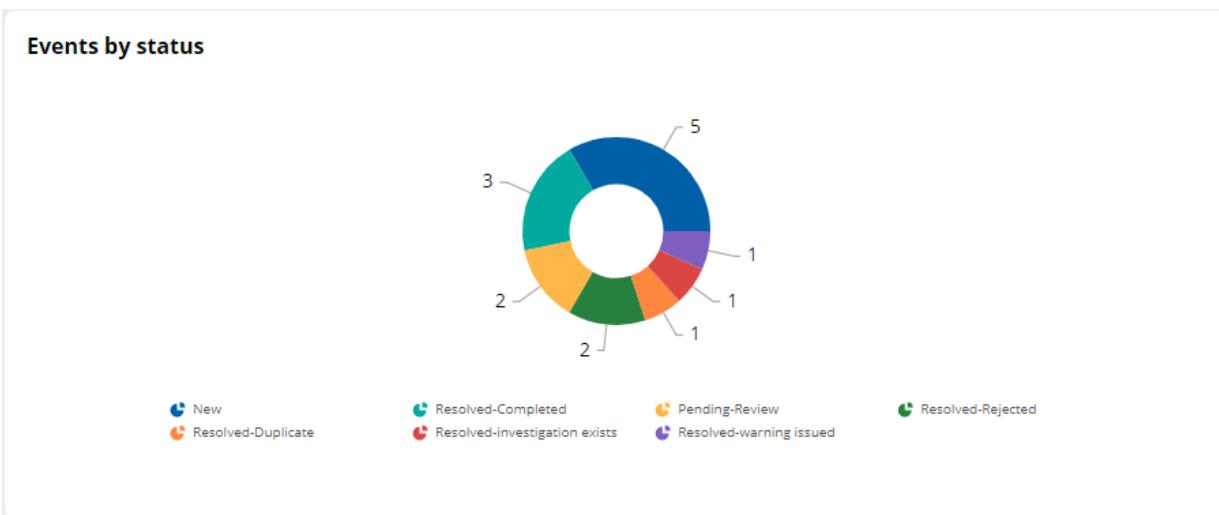
### Events by top 10 locations



### Events Report

To make changes open “EventsByLocation”(CL: *PegaPS-Work-Event-Initiate*) section and “EventsByLocation” (CL: *PegaPS-Work-Event-Initiate*) report definition.

- Event by Status: The following is a pie chart representing event cases against their status.



### Events report by status

To make changes, open “EventsByStatus”(CL: PegaPS-Work-Event-Initiate) section and “EventsByStatus” (CL: PegaPS-Work-Event-Initiate) report definition.



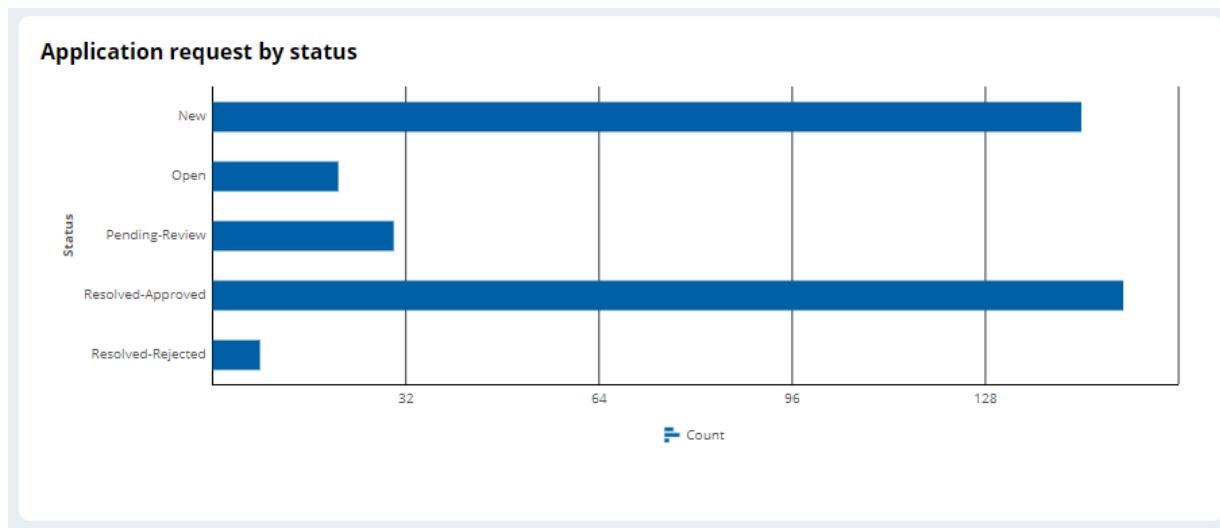
**Note:** The above report excludes resolved cancelled status.

## Reports on Application request

Users logging into PGPCosmos and ICMComos applications have reports on application requests in dashboard.

The following are the reports:

- Application Request by Status: The following Bar graph represents the count of applications based on their status.



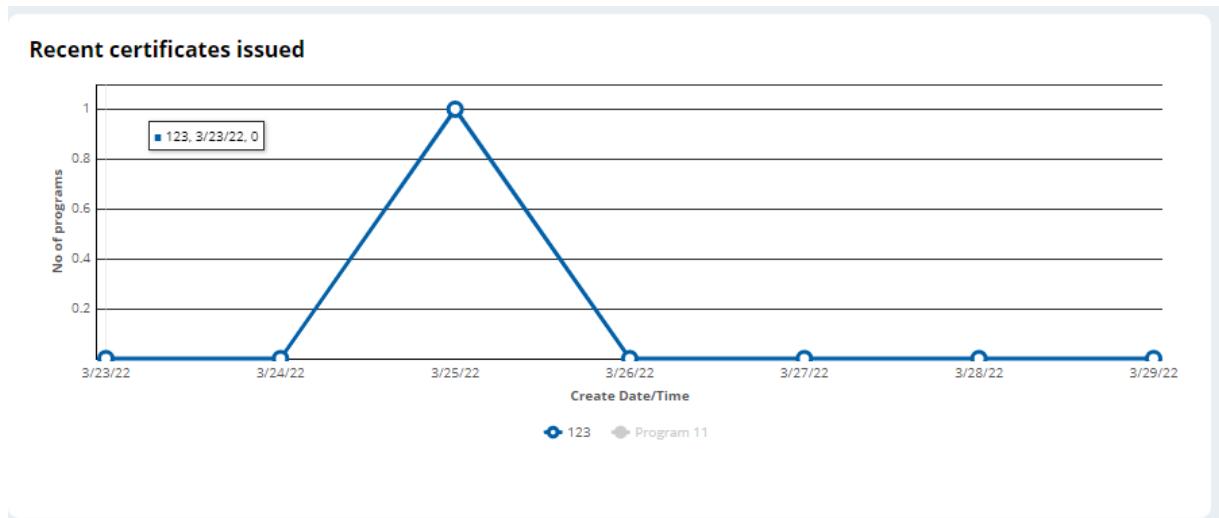
*Application request by status*

To make changes open “AppRequestByStatus”(CL: PegaPS-Work-ApplicationRequest) section and “AppRequestByStatus” (CL: PegaPS-Work-ApplicationRequest) report definition.



**Note:** The above report excludes resolved cancelled status.

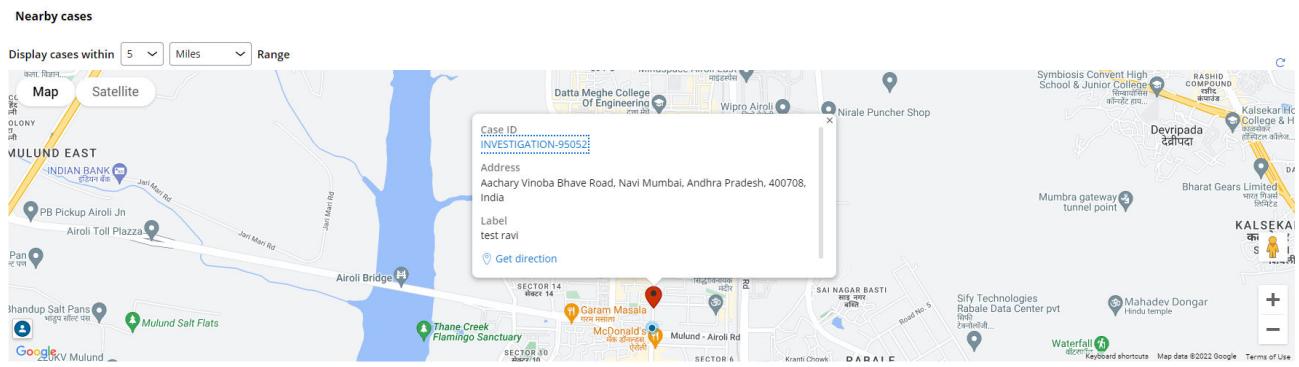
- Recent certificates issued: The following line graphs show different certificates approved in the last seven days.



*Certificates issued recently*

## Nearby cases chart

This chart provides the information on cases within the given range of distance.



*Nearby cases chart*

## Add a new report/chart to dashboard

To add a new report/chart to dashboard, follow these steps:

1. Open pyDefaultUserDashboard and add a step.
2. Set action as Update Page.
3. In Target, give `.pySlots(number).pyWidgets(number)` and set the following values as shown in example

▼ • 12	When	AccessGroup.pyAccessGroup==D_ICMApplicationContext.BusinessUnit	▼			
▼ • 12.1	Update Page	.pySlots(2).pyWidgets(1)	▼			
• 12.1.1	Set	.pySectionName	▼	equal to	"InvestigationByCategory"	▼ Select values +
• 12.1.2	Set	.pyClassName	▼	equal to	D_ICMApplicationContext.ClassNameValueGroup(W)	▼ Select values +
• 12.1.3	Set	.pyWidgetOptionsSectionName	▼	equal to	""	▼ Select values +
• 12.1.4	Set	.pyLabel	▼	equal to	"Investigation by category"	▼ Select values +
• 12.1.5	Set	.pyDescription	▼	equal to	"Investigation by category"	▼ Select values +
• 12.1.6	Set	.pySelected	▼	equal to	false	▼ Select values +
• 12.1.7	Set	.pyWidgetIsMasked	▼	equal to	false	▼ Select values +
▼ • 12.1.8	Update Page	.pyWidget	▼			
• 12.1.8.1	Set	Param.ClassName	▼	equal to	D_ICMApplicationContext.ClassNameValueGroup(W)	▼ Select values +
• 12.1.8.2	Apply Data Transform	SetObjClass	▼			

*Add report/chart*

## Edit labels of a chart

To edit labels of a chart, follow these steps:

1. Open the section with live UI and open the chart cell properties.
2. To change the Title go to General tab, under chart section click Edit Title.
3. To change axis labels, go to Axis tab and change Axis Label.

## Search program Landing Page

The landing page can be accessed from the left navigation pane. Programs/Licenses can be searched using details such as category, subcategory, name of program/License and entity type, entity name. The ID and Entity Name columns are provided as links to view details if required.

- **Adding a search field**
- **Adding field in search results**
- **Adding field to Search criteria**

## Adding a search field

To add a search field, complete the following steps:

1. Open D\_SearchProgramListdata page and add a parameter under the Parameters tab.
2. In Data Sources, open all the reports included as source.
3. In each report add parameter and update the filter conditions accordingly.

## Adding field in search results

To add a field in the search results, complete the following steps:

1. Open D\_SearchProgramList data page, in Data Sources open all the reports included as source.

2. In query tab of each report, add field as a column to display in the results.
3. Open ""SearchProgramTable"(CL:PegaPS) section, Add a column to the table and include the field.

## Adding field to Search criteria

To modify the search criteria open "SearchProgramTableHeader"(CL-PegaPS) and add or deletes fields as required.

## Manage associations

Use the Manage associations case to create or delete associations between entity types, such as Person, Business, Vehicle, Item, or Facility.

You can launch the Associations case from User Portal by performing either of the following actions:

- On Create menu, click Associations link.
- On Entity view screen or from case instance, in the Associations utility widget, click Gear icon.
- [\*\*Manage associations case architecture\*\*](#)
- [\*\*Association validations\*\*](#)
- [\*\*Adding a data element\*\*](#)
- [\*\*Adding an entity type\*\*](#)
- [\*\*Auditing an Association\*\*](#)
- [\*\*Extend Audit History for an Association\*\*](#)

## Manage associations case architecture

All associations are saved to and are retrieved from the Link database table (*PegaPS-Data-Link*). When associations are modified, the Link database table is also updated accordingly. The LinkedFromID is the primary ID from where the association is created for the first time. The LinkedToID is the ID to which the association is mapped.

LinkedToID is the ID to which the association is mapped. Associations are retrieved from the D\_AssociationListDetails data page which inturn invokes D\_AssociationList. The following are passed as parameters to the data page EntityID, EntityType and built-on application CSV.

- When Manage associations case is launched from the Entity view screen, the EntityID is the preset.
- When Manage associations case is launched from the User portal by using the Associations link on the Create menu, select Manage associations for field.

The D\_AssociationList is sourced from the *PegaPS-Data-Link* (Link) database table by using the EntityAssociations report definition. This report definition excludes entries from the Link table that have pyMemo set as Associate, Bid, Branch, Experience, Reporter, or Training that represents associations for other widgets of the entity view that are skipped in the Manage associations case to avoid redundancy.

FetchAssociationListDetails (Code-Pega-List) is the response data transform that is run on D\_AssociationList. This iterates through the results from the Link table and forms Primary.pxResults, which is of the *PegaPS-Data-Association*. This data transform has the logic to set the LinkedFromID, LinkedToID, LinkedClassFrom, LinkedClassTo, Role, IsReverseRole, and Entity Image data class fields, that are defined in the *PegaPS-Data-Association* class or its inheritance class to support associations.

Different categories of roles are maintained in the Option map (*PegaPS-Data-Config-OptionMap*) table, such as for example, PersonPerson, PersonBusiness, and

BusinessFacility. This table holds all of the possible roles between the Person to Person, Person to Business, and other associations.

Reverse roles are defined in the AssociationsRoleMapping and ICMAssociationsRoleMapping (*PegaPS-Data-Association*) decision tables for PGP and ICM layer. The FetchAssociationListDetails data transform uses this decision table to retrieve the reverse role if the associations are to be displayed in the context of the LinkedToID but not in the context from where the association is created.

## Association validations

There are certain validations in place for the associations that you can create by using the Associations case.

- You can add a new person, vehicle, or facility to your system when you create an association but you cannot create a business, as creating this entity involves a review process.
- You can associate an entity or case with another entity type or case type, including its own type. For example, you can associate a person (Person-1) with another person (Person-2). You can also associate a person (Person-1) with a business (Business-1).
- You can associate an entity or case with another entity or a case more than once with the same role. However, it cannot have a similar association within the same time period. For example, you can associate a person (Person-1) with another person (Person-2) with role as “Driver” multiple times, but with a different time period.

To update or add new validations, update the CheckDuplicateAssociation (*PegaPS-Work-Manage-Associate*) activity and the ManageAllAssociations (*PegaPS-Work-Manage-Associate*) validate rule.

For all associations, the context from where the association is created for the first time is important for displaying the relationship or role of the association. For example, when you create an association with a person (Person-1) and a business

(Business-1) as "Employer," then in the context of Business-1, Person-1 has an association with Business-1 as an "Employee."

## Adding a data element

If there is a requirement to capture a new data element, such as the association place, in addition to the relationship and the time period of the association, then complete the following steps.

1. Add a property to the *PegaPS-Data-Association* class, for example, Place.
2. Add a property to the *PegaPS-Data-Link* class, for example, Place.
3. Add a column to the database table that is mapped to the *PegaPS-Data-Link* class.
4. Fetch the property in the EntityAssociations report definition.
5. Add a mapping to the property in FetchAssociationListDetails.
6. Add the new property to the AssociationList (*PegaPS-Work-Manage-Associate*) section.
7. Add the new property to the SaveAllAssociates (*PegaPS-Work-Manage-Associate*) activity and SaveLink (*PegaPS*) activity

## Adding an entity type

The Manage associations case supports all five existing entity types, Person, Business, Facility, Vehicle, and Item in PGP and Investigation, Event, and Subject case types in ICM. If there is a requirement to include a new entity type (for example, Land) that was created in an application, then complete the following steps. Note: This example uses "Land" as the new entity type. The entity should be replaced wherever applicable with the new entity type that you are adding.

1. In the Option map (*PegaPS-Data-Config-OptionMap*) data type, in the field **FromAssocType**, add a new entry for Land.
2. In the Option map (*PegaPS-Data-Config-OptionMap*) data type, for the new field value **LandAssocType**, add entries for Person, Business, Facility, and Vehicle.
3. Flush **D\_OptionMapList** data page and **D\_PGPOptionMapList** for PGP, **D\_ICMOptionMapList** for ICM.

4. Update the **D\_EntityList** data page to add logic for land to fetch the list of Land details.
5. Update the **PreFromAssociateSearch** (*PegaPS-Work*) data transform to add logic for the new entity type. Refer to the existing steps added for a Person Entity in the **PreFromAssociateSearch** data transform and add steps in data transform for Land entity accordingly.
6. Create a **FromAssociateSearch** section for the new entity to add search criteria and results.
7. Update the **PreAddAllAssociates** and **ResetAddAssociate** data transforms to add logic for Land. Refer to the existing steps added for a Person Entity in the data transforms and add steps for Land entity accordingly.
8. Create AssociateLand section and Update the AddAllAssociates section to include the AssociateLand section and add a visibility condition based on the AssociateType.
9. Create a data transform for initializing a new Land page, which is referenced when a new Land is created and added as an associate.
10. Create a new validate rule for validating Land Entity related fields and update ValidateAssociations validate rule to call the new validate rule.
11. Update PostAddAllAssociates activity to add logic for Land. Refer to the existing steps related to Person Entity and add the same for Land accordingly.
12. Update CheckduplicateAssociation activity to add logic for Land.
13. Update SaveAllAssociates activity to add logic for Land.

## Auditing an Association

Pega Government Platform provides support to Audit History for all the modifications done as part of an Association case. Below are the scenarios where case history is updated for Association case.

1. Adding new associations.
2. Modifying the existing association details like role, change in time period.
3. Removing the associations.

## Extend Audit History for an Association

To extend audit history for a new field, update PropertiesToCompare (CL:PegaPS-Data-Association) data transform to set a new field.

For example, to add audit history for the new field 'Place', open the PropertiesToCompare data transform and add another step to set the new field 'Place'.

## Time Management

Time Management application provides a generalized time management system to track and report time for case workers that is consistent with applicable standards and guidelines.

Users use this application to capture, submit time entries for approval, re-submit rejected time entries on a daily basis. Users have multiple opportunities to record their time throughout the workday or they can enter time for the entire day and submit for approval as a part of their end-of-day activities.

The Managers can view the time entries submitted by their workgroup and they can Approve or Reject the time entries.

Data tables used in Time Management application are as follows:

- The Project data table stores and lists all the projects details available in the system. This table contains information about projects such as Project ID, Project name, Project details, Project approver, and so on.
- The Time code table contains information about time codes such as name, description, Approver, status, and so on.

New Time entry

Day\*  
7/1/2022

Copy projects, activities and references from latest entry

Hours

Project*	Activity*	Reference	Start	End	Hours	Status	Action
						Not submitted	

+ Add

Total hours 0.0  
Time sheet due 7/1/22 11:59 PM  
Case history Touchpoint report

[Cancel](#) [Submit](#)

*Time entry*

## Manage Time

Time Management also has a Manage time landing page in which users can view time entry logs for a given time period; a manager can view all time entry logs for a team.

Manage time

MY TIMESHEET TEAM TIMESHEET

Start Day\* 6/24/2022 End Day\* 7/1/2022 Project All Activity All Reference All Clear all filters

Total tracked 23.9 hours Total approved 2.0 hours Total pending approval 17.9 hours Total rejected hours 4.0 hours

Group Fields Density Refresh

Date	Project	Activity	Reference	Status	Hours
Jun 29, 2022	Case Worker	Research	AP-95005	Submitted	1.2
Jun 29, 2022	Case Worker	Investigation		Submitted	1.9
Jun 29, 2022	Investigation Project	Collecting evidence		Submitted	0.8
Jul 1, 2022	Case Wo	Surveillance		Rejected	4.0
Jun 30, 2022	Case Worker	Research		Approved	2.0
Jul 1, 2022	Case Worker	Research	TE-30007	Submitted	1.0
Jul 1, 2022	Case Worker	Research		Submitted	4.0
Jul 1, 2022	Case Worker	Surveillance Observation		Submitted	6.0
Jul 1, 2022	Case Worker	Research	TE-30016	Submitted	3.0

*Manager view - Manage time*

Time management also provides restricted access to time entries, which is enforced by attribute based access control (ABAC) rules:

1. Users access their own entries.

2. Managers access entries for themselves and from case workers in their work group.
3. Administrators view all entries.

- **Adding new fields to project tables**
- **Adding, deleting, or updating records in a project table**
- **Adding new fields to Time code data objects**
- **Adding, deleting, or updating records in Time code data objects**
- **Adding new fields to the time entry view**
- **Adding new fields to time entry log data objects**
- **Changing the due time of a time sheet**
- **Time entry flow**

## Adding new fields to project tables

Add a new field to the project table in the Time Management application.

1. In the navigation pane of App Studio, click Data > Data objects and integrations.
2. In the list of data objects, click Project table.

← Data objects and integrations

Phone

Plan	Plan SIMULATED	PlanList SIMULATED
Portal	Config list	Portal config list
Program config	Active program list SIMULATED	Program config list
Project table	List project table	Project
Qualification	<small>Previous answers in a category</small> Project table data object	

3. On the Data model tab, check all the existing fields.
4. Click Add field.
5. Click Save.

Name	ID	Type	Options	Application Layer
Can persist	CanPersist	Boolean		Time management
Effective end date	EffectiveEndDate	Date & time		Time management
Effective start date	EffectiveStartDate	Date & time		Time management
Funds allocation	FundsAllocation	Currency		Time management
Globally unique ID	pyGUID	Text (single line)	Key ( autogenerated); Read-only	Pega Platform
Project active	Active	Boolean		Time management
Project approver	Approver	Text (single line)		Time management
Project details	Details	Text (paragraph)	Plain text	Time management
Project id	ProjectId	Text (single line)		Time management
Project name	Name	Text (single line)		Time management

*Data model*

## Adding, deleting, or updating records in a project table

Add, delete, or update records in project table:

1. In the navigation pane of App Studio, click Data > Data objects and integrations.
2. In the list of data objects, click Project table.

← Data objects and integrations

Phone

Plan	Plan SIMULATED	PlanList SIMULATED
Portal	Config list	Portal config list
Program config	Active program list SIMULATED	Program config list
Project table	List project table	Project
Qualification	<small>Previous answers in a category Project table data object</small>	

3. Click the Records tab.

Project id	Project name	Project details	Funds allocation	Effective start date	Effective end date	Project approver	Project active	
P005	Investigation Project			6/2/2022 9:41 AM	7/1/2022 9:41 AM	pudua	✓	
P004	Administrative		\$600,000.00	5/1/2022 5:46 PM	6/1/2022 5:47 PM	sriks1	✓	
P001	Case Worker		\$500,000.00	4/2/2022 1:02 AM	6/1/2023 12:02 AM	sravan	✓	
P002	Regular		\$200,000.00	6/1/2021 8:39 PM	6/30/2022 8:39 PM	TMAapprover	✓	
P003	Other					pudua		

*Records tab in of the project table*

4. Click on any row to update the project record.
5. Click the Delete icon to delete the project record.
6. Click Add record to add a new project.

## Adding new fields to Time code data objects

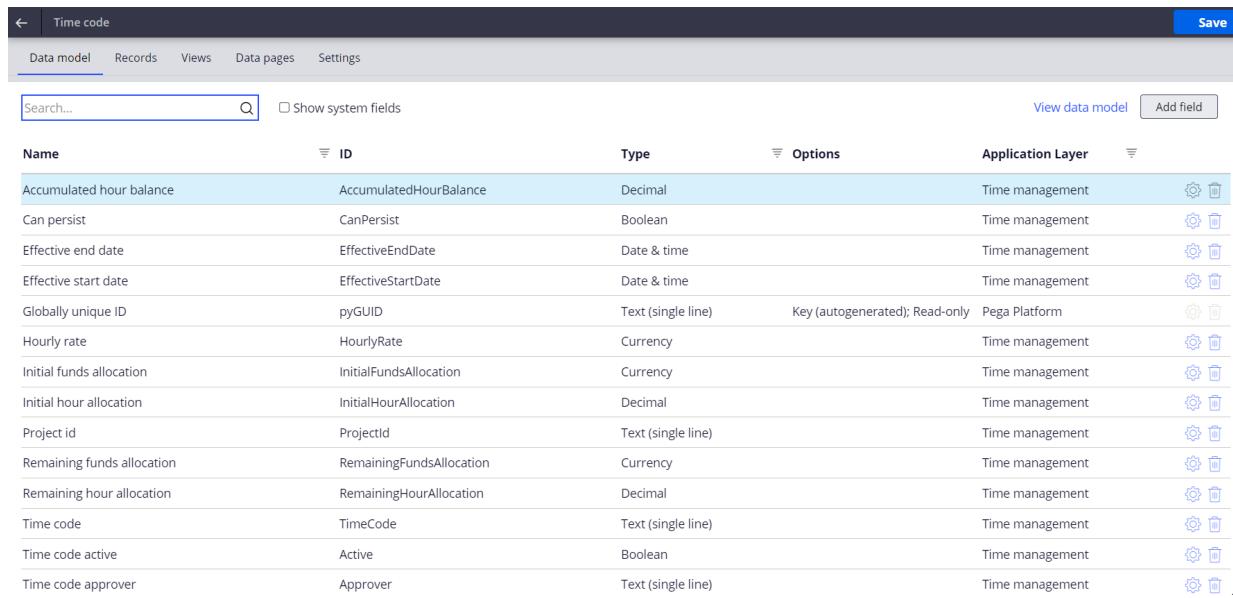
Add a new field to a Time code data object to xxx

1. In the navigation pane of App Studio, click Explore Data.
2. In the **Data objects and integrations** window, click the Time code data object.

Subject	ICMEntity_Editable		
Task	Task SIMULATED	Task list SIMULATED	
Time code	List time code	Time code	
Time entry log	Case specific logs	List time entry log	
Training	D_Training SIMULATED	PersonTrainingList SIMULATED	
Vehicle	Data vehicle SIMULATED	Fetch vehicle proximity SIMULATED	

*The Time code data object*

3. On the Data model tab, review the existing fields.
4. **Optional:** To add a new field, click Add field.



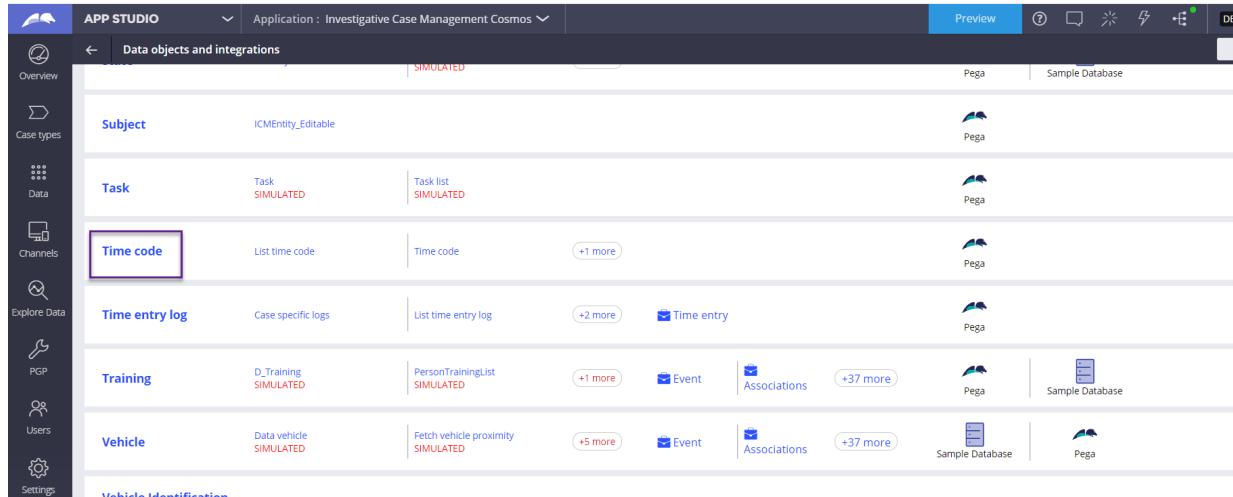
Name	ID	Type	Options	Application Layer
Accumulated hour balance	AccumulatedHourBalance	Decimal		Time management
Can persist	CanPersist	Boolean		Time management
Effective end date	EffectiveEndDate	Date & time		Time management
Effective start date	EffectiveStartDate	Date & time		Time management
Globally unique ID	pyGUID	Text (single line)	Key (autogenerated); Read-only	Pega Platform
Hourly rate	HourlyRate	Currency		Time management
Initial funds allocation	InitialFundsAllocation	Currency		Time management
Initial hour allocation	InitialHourAllocation	Decimal		Time management
Project id	ProjectId	Text (single line)		Time management
Remaining funds allocation	RemainingFundsAllocation	Currency		Time management
Remaining hour allocation	RemainingHourAllocation	Decimal		Time management
Time code	TimeCode	Text (single line)		Time management
Time code active	Active	Boolean		Time management
Time code approver	Approver	Text (single line)		Time management

*The Data model tab of the Time code window*

## Adding, deleting, or updating records in Time code data objects

Add, delete, or update records in Time code data objects to xxx.

1. In the navigation pane of App Studio, click Explore Data.



Subject	Description	Associated Objects
ICMEntity_Editable		Pega
SIMULATED		
Task	SIMULATED	Pega
SIMULATED		
Time code	List time code	Time code +1 more
		Pega
Time entry log	Case specific logs	List time entry log +2 more Time entry
		Pega
Training	D_Training SIMULATED	PersonTrainingList SIMULATED +1 more Event Associations +37 more
		Pega Sample Database
Vehicle	Data vehicle SIMULATED	Fetch vehicle proximity SIMULATED +5 more Event Associations +37 more
		Sample Database Pega
Vehicle Identification		

*Time code data object*

2. In the **Data objects and integrations** window, click the Records tab to view the time codes that are available in the system.

project id	Time code	Time code description	Time code approver	Hourly rate	Initial funds allocation	Remaining funds allocation	Initial hour allocation	Remaining hour allocation
P002	T009	Leave	sravan					
P001	T001	Surveillance Observation	pudua	\$50.00	\$100,000.00	\$91,800.00	2000.0	1836.0
P003	T008	Conduct Interview	ankig					
P002	T005	Training	sravan	\$100.00	\$200,000.00	\$10,000.00	1000.0	500.0
P002	T010	Observations	sriks1					
P001	T002	Training	ramas13					
P001	T003	Research	sriks1	\$50.00	\$100,000.00	\$73,150.00	2000.0	1463.0
P001	T004	Investigation	TMUser2	\$75.00	\$300,000.00	\$259,200.00	4000.0	3456.0
P005	T007	Collecting evidence	pudua					
P005	T100	Report	tiruk					

*Time code records*

3. **Optional:** To update a time code record, click the row of the record that you want to update.
4. **Optional:** To delete a time code record, click the Delete icon.
5. **Optional:** To add a new time code, at the end of the list click Add record.

## Adding new fields to the time entry view

Add a new field to the time entry view xxx.:

1. In the navigation pane of App Studio, click Explore Data.
2. In the **Case types** window, click Time entry.

Case types	Type	Referenced data objects
Time entry	Standard	Time entry log

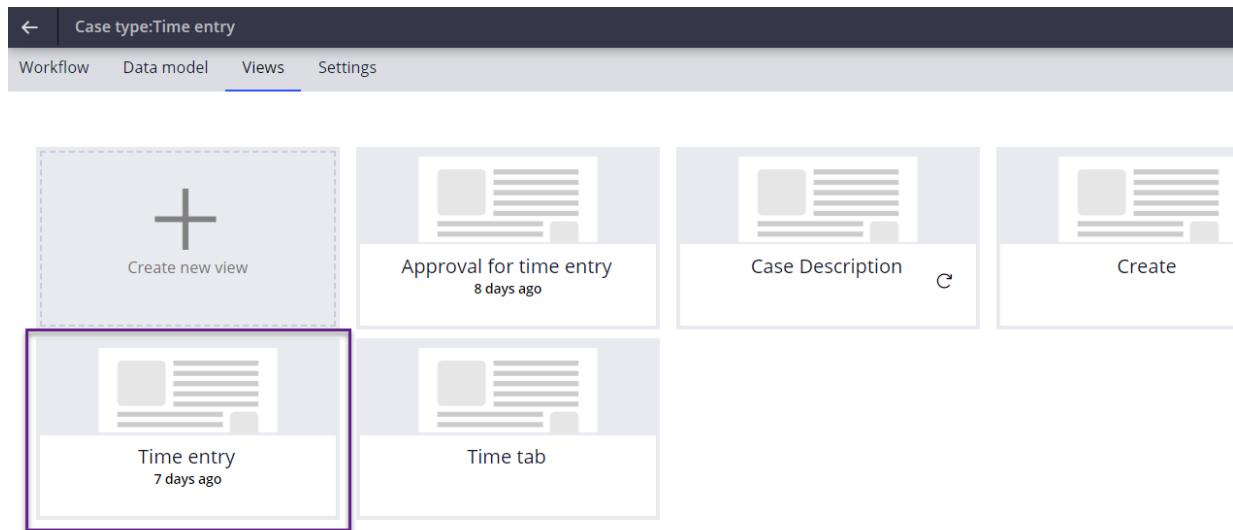
*Time entry case type*

3. On the Data model tab, review the existing fields.
4. **Optional:** To add a new field, at the end of the list click Add field.

Name	ID	Type	Options	Application Layer
Case ID	pyID	Text (single line)	Key; Read-only	Pega Platform
Comment	Comments	Text (paragraph)	Plain text	Time management
Description	pyDescription	Text (paragraph)	Plain text	Pega Platform
Duplicate Case Reference	pyDuplicateCaseReference	Case reference	Case	Pega Platform
Label	pyLabel	Text (single line)		Pega Platform
Operator id	OperatorID	User reference		Time management
Reviewer action	Action	Picklist	Radio-buttons	Time management
Time entry logs count	TimeEntryLogsCount	Integer	Calculated; Read-only	Time management
Time sheet due	TimeSheetDue	Date & time		Time management
Total hours	TotalHours	Decimal	Calculated; Read-only	Time management
Work status	pyStatusWork	Text (single line)		Pega Platform

*Time entry - Data model tab*

## 5. Click Views > Time entry.

*Time entry - the Views tab*

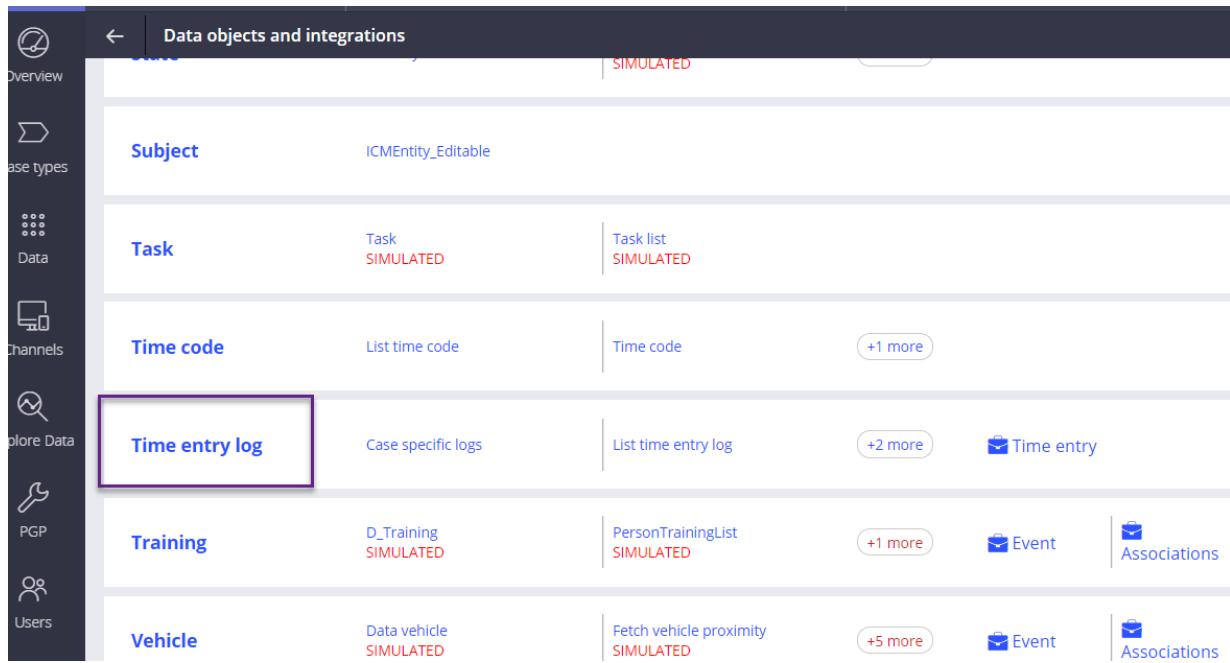
## 6. Search for and add the new field to the view.

# Adding new fields to time entry log data objects

Add a new field to time entry log data objects to xxx.

## 1. In the navigation pane of App Studio, click Explore Data.

2. In the **Data objects and integrations** window, click the Time entry log data object.



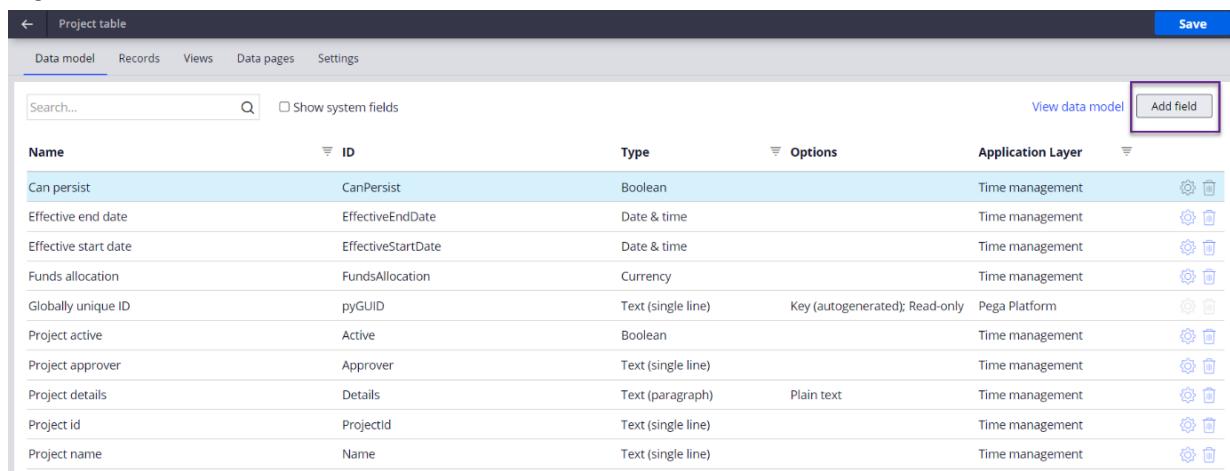
The screenshot shows the 'Data objects and integrations' window. On the left is a sidebar with icons for Overview, Base types, Data, Channels, Explore Data, PGP, and Users. The main area lists several data objects:

- Subject**: ICMEntity\_Editable
- Task**: Task SIMULATED
- Time code**: List time code
- Time entry log**: Case specific logs (highlighted with a purple box)
- Training**: D\_Training SIMULATED
- Vehicle**: Data vehicle SIMULATED

For each object, there are details like type, status (e.g., SIMULATED), and associated components like Task list, Time code, and Event.

The Time entry log data object

3. On the Data model tab, review the existing fields.  
 4. **Optional:** To add a new field, click Add field.



The screenshot shows the 'Project table' - 'Data model' tab. It displays a table of fields:

Name	ID	Type	Options	Application Layer
Can persist	CanPersist	Boolean		Time management
Effective end date	EffectiveEndDate	Date & time		Time management
Effective start date	EffectiveStartDate	Date & time		Time management
Funds allocation	FundsAllocation	Currency		Time management
Globally unique ID	pyGUID	Text (single line)	Key (autogenerated); Read-only	Pega Platform
Project active	Active	Boolean		Time management
Project approver	Approver	Text (single line)		Time management
Project details	Details	Text (paragraph)	Plain text	Time management
Project id	ProjectId	Text (single line)		Time management
Project name	Name	Text (single line)		Time management

At the top right of the table, there is a 'Save' button and a 'View data model' button. Below the table, there is a 'Search...' input field and a 'Show system fields' checkbox.

Time entry log - Data model tab

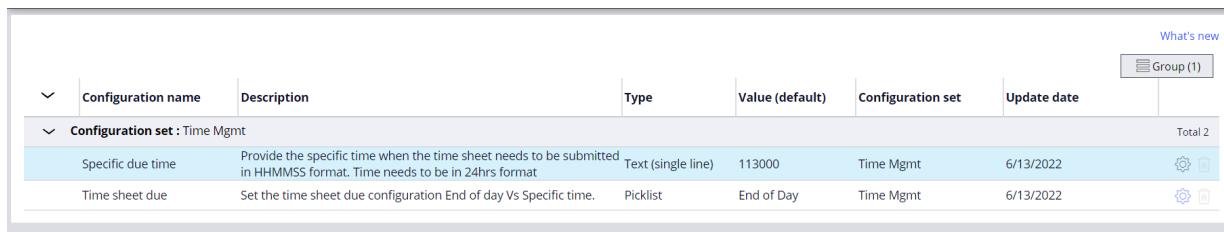
5. Switch to Dev Studio.  
 6. In Dev Studio, save the TimeEntry\_TimeEntryLogs (CL:PS-TimeMgmt-Work-TimeEntry) section and add the newly added field as a column to the table layouts.

7. Repeat step 6 to save the following sections into the implementation layer and add a new field as a column to the table layout:
  - *ApprovalTimeEntryLog*
  - *Approval\_TimeEntryLogs*
8. Add the field that you added in step 7 as a column in the following report definitions class of PS-TimeMgmt-Data-TimeEntryLog:
  - *TimeEntryLogList*
  - *LatestTimeEntryLogList*
  - *MyTimeLogListLP*
  - *CaseTimeEntryLogList*

## Changing the due time of a time sheet

Configure the Time sheet due setting to change the due time for a time sheet. The default end of the day for a time sheet is 11:59 PM. This is called the due time. If a time entry case is submitted after the due time, the system displays a validation error and the user can not submit the case again.

1. In the navigation pane of App Studio, click Settings > Configurations.
2. Click the Settings icon at the end of the Time sheet due row.
3. Select Specific time from the list and then click Submit.
4. Click the Settings icon at the end of the Specific due time row.
5. Enter a new due time as HHMMSS, in 24-hour format, for when users must submit the time sheet.



Configuration name	Description	Type	Value (default)	Configuration set	Update date	Actions
<b>Configuration set : Time Mgmt</b>						
Specific due time	Provide the specific time when the time sheet needs to be submitted in HHMMSS format. Time needs to be in 24hrs format	Text (single line)	113000	Time Mgmt	6/13/2022	
Time sheet due	Set the time sheet due configuration End of day Vs Specific time.	Picklist	End of Day	Time Mgmt	6/13/2022	

*update timesheet*

## Time entry flow

This feature allows users to log time against a case for the current day. The time logs are reviewed by manager or supervisor and they can approve or reject the time log.

The flow can be reused in such a way that users can initiate time entry log from any case type. As it is initiated from a case the reference details are auto populated.



*time entry flow*

## View time in case

This feature allows user to view the summary chart and reports on time entry logs for a particular case.

## Time entry in Investigation

In investigation case users can log time from Action > Add work and the reference details are auto populated. The View time in case is also included to the case but only the investigation owner and manager/supervisor can view it.

## How to add Time Entry to a case type

Open case type in app studio, Optional actions > add a new process to a stage > add step > More > Processes > Time Entry

## How to add a Time tab (View time in case)

Open "pyCaseMainInner" section rule of that class, under Main content add a section control with section name "TimeWrapper" with page context as "Use current page context."

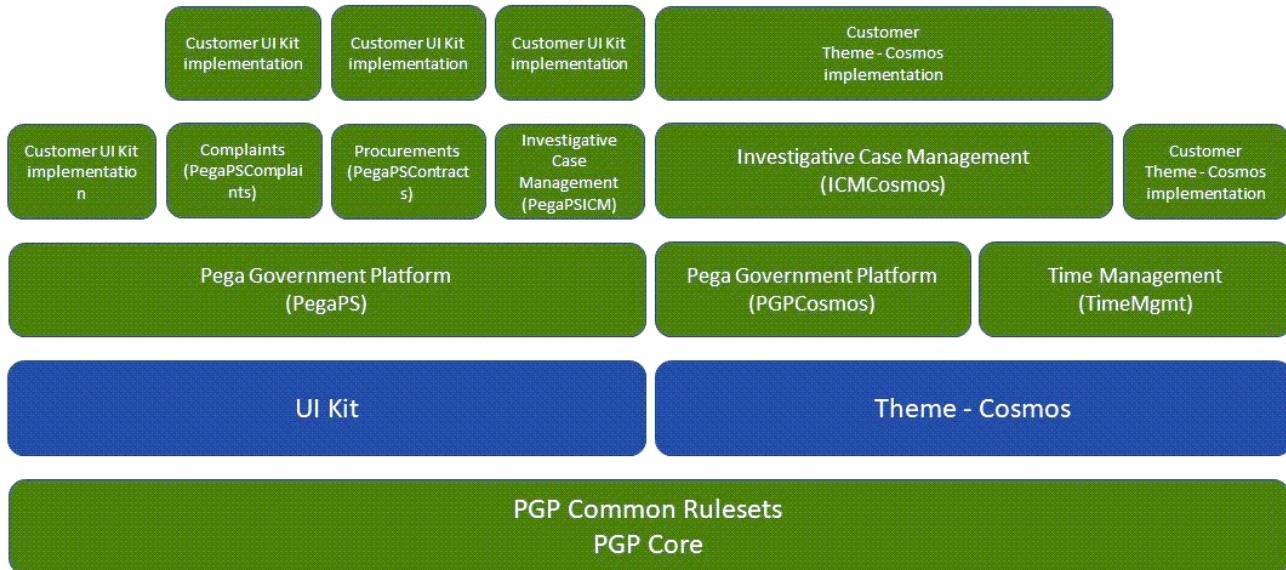
# Appendix A: Pega Government Platform Application overview

## Business issue

Government applications require modern web behaviors, consistent with the mainstream web applications to minimize operator training and shorten their learning curve. The Pega UI Kit Design System uses traditional web mechanics like iFrame; full page refreshes and avoids modern browser capabilities, for example, back button and tabs.

## Pega Government Platform solution

In addition to the traditional PegaGP application stack based on the UI Kit design system, Pega Government Platform 8.6 offers the PegaGPCosmos and the new PGPICMCosmos application stacks based on the new Cosmos design system of Pega.



*New design system of Pega*

## Pega Government Platform 8.6 solution scope

From release 8.6, the PGP Core application features are available in the PegaGPCosmos application stack and the Investigative Case Management features are available in the PGPICMCosmos application stack.

## Customer use

In App Studio, use the New Application Wizard and select PegaGPCosmos as the built-on application type for a new implementation layer application. Follow the design guidance available on the [Pega Cosmos design system website](#).

## Key topics

### Transition from UI Kit

There is no automated path to transition from the UI Kit design system to the Theme Cosmos design system. A UI Kit based application must be manually migrated to the new design system through developer effort. If your Pega 8.4 or earlier application does not override or extend Platform UI rules and uses Design

Templates and Views to implement its UI, it easily transitions to the Cosmos design system. For more information, see the Cosmos [Development Articles](#).

## Differences in design systems

UI-Kit	Cosmos
In UI Kit you can use Dev Studio sections with the full section editor.	In Cosmos, you can use App Studio friendly design templates and views.
In UI Kit and Pega 8.4 and earlier applications, you can override and extend unconfirmed platform rules that do not meet your implementation layer requirements.	In Cosmos applications, you configure platform rules to meet your requirements and rigorously avoid overriding them.

## Development approach

Starting in Pega Platform version 8.5, a case types can have an initial stage: Create Stage. Cosmos uses this new stage to control the initial presentation dialog when a case type creates new work. The existing case types that are migrated from the UI-Kit to Cosmos must have this Create Stage manually added.

When overriding a Platform UI rule is unavoidable, you must keep a track of the rule. For example, in separate Platform-overrides ruleset, and detail the reason(s) this override could not be avoided in the rule Usage. With each subsequent release of the Pega Platform and Pega Government Platform reevaluate the overridden rules and seek to withdraw them.

## PGP Theme Cosmos developer notes

### PGP Core Application stack comparison

Traditional PegaGP:8 applicationstack	New PegaGPCosmos:8 applicationstack
Built-On UIKit:15.01	Built-On Theme-Cosmos:02.01
Built-On PegaRULES:8	Built-On PegaRULES:8

<b>Traditional PegaGP:8 applicationstack</b>	<b>New PegaGPCosmos:8 applicationstack</b>
	PGPCosmos:08-06
PGPOVERRIDE:08-06	PGPCosmosOverride:08-06
PegaPSPortals:08-06	PegaPSPortals:08-06
PegaPSUserServices:08-06	PegaPSUserServices:08-06
PegaPS:08-06	PegaPS:08-06
PegaPSInt:08-06	PegaPSInt:08-06
PegaPSUI:08-06	PegaPSUI:08-06
PegaPSIntConnectors:08-06	PegaPSIntConnectors:08-06
PegaPSConnectors:08-06	PegaPSConnectors:08-06
PegaPSGDPR:08-06	PegaPSGDPR:08-06

## ICM Core Application Stack Comparison

<b>Traditional PGPICM:8 applicationstack</b>	<b>New PGPICMCosmos:8 applicationstack</b>
Built-On PegaGP:8	PegaGPCosmos:8
<b>Traditional PGPICM:8 applicationstack</b>	<b>New PGPICMCosmos:8 applicationstack</b>
	ICMCosmos:08-06
ICMOVERRIDE:08-06	ICMCosmosOverride:08-06
PegaPSICM:08-06	PegaPSICM:08-06
PegaPSICMINT:08-06	PegaICMINT:08-06
PegaPSICMPORTALS:08-06	PegaPSPortals:08-06

In Pega Government Platform 8.6, two application rules are provided for PGP Core, and two application rules are provided for ICM: one application rule

supports building traditional UI Kit-based applications and the other supports building new Cosmos-based applications.

**PegaGP:8** – provides the UI-Kit based ruleset stack that is consistent and backward compatible with previous versions of Pega Government Platform.

**PegaGPCosmos:8** – provides the same feature set as PegaGP but with the common Cosmos-themed user interface rules.

**PGPICM:8** – provides the UI Kit based ruleset stack that is consistent and backward compatible with previous versions of PGP Investigative Case Management.

**PGPICMCosmos:8** – provides the same feature set as PGPICM but with the common Cosmos-themed user interface rules.

Both application stacks share the common rulesets for Pega Government Platform data model, PGP connector, and PGP work processing rules, also known as, PGP Core.

The Cosmos design application contains two additional rulesets:

**<app>Cosmos:08-06** – this ruleset contains user interface rules that Pega Government Platform has overridden (copied) from the UI Kit design system and extended for specific purposes within Pega Government Platform.

**<app>CosmosOverride:08-06** – this ruleset contains user interface rules overridden from Theme Cosmos to follow the implementation guidance from the Pega User Experience team.

## Specific overrides

Specific UI overrides interfere with future UI upgrades and should be avoided. The rules in the PGPCosmosOverride ruleset are purposefully isolated from the Pega Government Platform application rules. In the future releases, Pega Platform and the

common Cosmos design include the features that the rules in the PGPCosmosOverride ruleset provide.

### **pyWorkCommonActions navigation rule**

This navigation rule was overridden on Pega Government Platform entity view case types to customize the list of actions available. Pega Platform 8 does not offer a mechanism to configure and extend a navigation rule except to override.

Rule type	Rule name	From class	To class
Navigation	pyWorkCommonActions	Work-	PegaPS-Work-Entity
Navigation	pyWorkCommonActions	Work-	PegaPS-Work-Entity-Business-View
Navigation	pyWorkCommonActions	Work-	PegaPS-Work-Entity-Facility-View
Navigation	pyWorkCommonActions	Work-	PegaPS-Work-Entity-Household-View
Navigation	pyWorkCommonActions	Work-	PegaPS-Work-Entity-Person-View
Navigation	pyWorkCommonActions	Work-	PegaPS-Work-Entity-Vehicle-View

### **pyCaseMainInner section rule**

This section rule was overridden to add Pega Government Platform entity-specific tabs on the left panel and Pega Government Platform utility widgets to the right panel. Pega Platform 8 does not offer a mechanism to configure included widgets on a section rule except to override.

Rule type	Rule name	From class	To class
Section	pyCaseMainInner	Work-	PegaPS-Work
Section	pyCaseMainInner	Work-	PegaPS-Work-Entity
Section	pyCaseMainInner	Work-	PegaPS-Work-Entity-Business-View
Section	pyCaseMainInner	Work-	PegaPS-Work-Entity-Facility-View
Section	pyCaseMainInner	Work-	PegaPS-Work-Entity-Household-View
Section	pyCaseMainInner	Work-	PegaPS-Work-Entity-Person-View
Section	pyCaseMainInner	Work-	PegaPS-Work-Entity-Vehicle-View

### pyCaseConfirm section rule

This section rule was overridden to provide a Pega Government Platform specialized Toast Message in the case completion view.

Rule type	Rule name	From class	To class
Section	pyCaseConfirm	Work-	PegaPS-Work
Section	pyCaseConfirm	Work-	PegaPS-Work-Entity
Section	pyCaseConfirm	Work-	PegaPS-Work-Manage

### CaseActionHeader section rule

This section rule was overridden to add an Edit button for entity view case types, and to add a View More button for entity add case types.

Rule type	Rule name	From class	To class
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Business
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Business-View
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Facility
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Facility-View
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Household
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Household-View
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Person
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Person-View
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Vehicle
Section	CaseActionHeader	Work-	PegaPS-Work-Entity-Vehicle-View

## pyRefresh navigation rule

This navigation rule was overridden on entity work classes to invoke the Pega Government Platform *PegaPS-Work ClearHomeDataPages* data transform rule passing the *EntityType* and *EntityID* parameters which causes the removal of data pages related to the current entity before the default Refresh action occurs.

Rule type	Rule name	From class	To class
Navigation	pyRefresh	Work-	PegaPS-Work-Entity-Business-View
Navigation	pyRefresh	Work-	PegaPS-Work-Entity-Facility-View
Navigation	pyRefresh	Work-	PegaPS-Work-Entity-Household-View
Navigation	pyRefresh	Work-	PegaPS-Work-Entity-Person-View
Navigation	pyRefresh	Work-	PegaPS-Work-Entity-Vehicle-View

### HomeMain section rule

This section rule was overridden and circumstanced for mobile to remove the pulse display on the Home Page of the Cosmos design User Portal.

Rule type	Rule name	From class	To class
Section	HomeMainBase	Data-Portal	Data-Portal
Section	HomeMainpxDevice = Phone	Data-Portal	Data-Portal

### CaseDetails section rule

This section rule is overridden for the Pega Government Platform Event case type to customize the Primary Case Data section details.

Rule type	Rule name	From class	To class
Section	CaseDetails	Work-	PegaPS-Work-Event

## WelcomeBanner section rule

This section rule was overridden to add Pega Government Platform specific welcome text to the Theme Cosmos welcome screen on the User Portal.

Rule type	Rule name	From class	To class
Section	WelcomeBanner	Data-Portal	Data-Portal

## pyDefaultUserDashboard data transform rule

This data transform rule was overridden in order to add Pega Government Platform Event status information to the Theme Cosmos User dashboard.

Rule type	Rule name	From class	To class
Data Transform	pyDefaultUserDashboard	System-User- Dashboard	System-User- Dashboard

# Appendix B: Creating new entities

## What are PGP entities?

PGP entities are a collection of robust data structures that model data elements that are commonly used in government processes. The PGP entities are persistent data classes that come pre-configured as a part of PGP.

- Person
- Business
- Vehicle
- Facility
- Household

Entity data structures serve two distinct purposes within PGP: first, Entity instances are persisted to a data store like any local data type or data table; second, Entity data classes are used as a type for embedded pages defined on the PGP work ancestor to support case processing.

## Persistent Data Store

An entity data class (Data- class) is managed (curated) in an independent data store (database table) by the *Add, Update, View and Search* case types (Work-classes).

## Case Type embedded page

An entity data class is also used as a type for embedded pages incorporated into the PGP work ancestor. These embedded pages are used by the PGP component processes to implement PGP features.

PGP Entities follow a pattern of Persistent Data curated by Business Processes. This pattern allows for the possibility that in a customer implementation the entity datastore may be mapped to customer resource that is external to Pega and PGP. In this pattern a persistent datastore descended from *PegaPS-Data-Entity* is added, updated, viewed, and searched by Case Types descended from *PegaPS-Work-Entity*.

## New entity overview

This guide will step through the process of extending the PGP Entity data structure adding a new entity called *Item* built in a PGP implementation layer.

In the following examples *PGPImp* is the organization name of the implementation application and *DemoPGP* is the name of the implementation application. The class *PGPImp-DemoPGP-Work* inherits (directed) from the *PegaPS-Work* class.

### Creating the persistent datastore

`PGPImp-DemoPGP-Data-Entity-Item`

- Properties relevant to the entity
- View(s) for editing, reviewing
- Data Pages for retrieving
- Optional: Case match rule to avoid duplicates, and validation rules for data

## Extending the work class properties

In the primary work class of the implementation application, *PGLmp-DemoPGP-Work*, add a page property of type *PGLmp-DemoPGP-Data-Entity-Item* to hold *Item* data during case management processing.

## Creating the data management case types

`PGLmp-DemoPGP-Work-Entity-Item-Add`

`PGLmp-DemoPGP-Work-Entity-Item-View`

`PGLmp-DemoPGP-Work-Entity-Item-Search`

`PGLmp-DemoPGP-Work-Entity-Item-Update`

## Enabling for App Studio

Use views and design templates to construct the user interface.

Catalog the new Entity's rules in Relevant Records so they will appear in the App Studio smart prompts.

Configure whether case types will appear on +Create nav.

## Development steps

While it is possible to do some of this work in App Studio, the approach discussed here is based on a Dev Studio experience. App Studio hides some development details, like class names and rulesets, to present a development environment that can focus on case types and business processes, but for this exercise we need full control over those details.

### Creating persistent datastore

Create a persistent data type and supporting rules for the new entity.

## Creating data type ancestor

Add an abstract data class to the implementation layer to leverage the Enterprise Class Structure pattern and provide extension points for reusable rules common to the Entity implementation layer data types.

1. To create a new class, click **Create > SysAdmin > Class**.
2. In the **Label** field, enter Entity.
3. In the **Class Name** box, enter or select PGPImp-DemoPGP-Data-Entity.
4. Click Create and open.
5. In the **Select class type** list, select Abstract.
6. In the **Settings** section, in the **Created in version** box, enter or select the current version.
7. In the **Class inheritance** section, in the **Parent class (Directed)** box, enter or select PegaPS-Data-Entity.
8. Click **Save**.

## Creating data type

Create the new *Item* data type and supporting rules.

1. In the navigation pane of Dev Studio, click **Data types**.
2. Click Options > Add data type.
3. In the **Add Data Type** dialog box, click **New Data Type**.
4. In the **Label** field, enter **Item**.
5. In the **Description** field, enter an appropriate description for the class.
6. Expand **Advanced** section.
7. In the **Parent class** box, enter or select PGPImp-DemoPGP-Data-Entity.
8. Click **Submit** to create the new data type and open it for editing
9. On the **Data model** tab, use the Add field button to add properties the *Item* data type
  - a. Add a property named EntityID of type Text.
  - b. Add a property pyID of type Text.
  - c. Continue adding properties as desired to define the Item data type.
  - d. Save the data type.

10. Switch to the **Sources** tab and click **Create a local source**.
  - a. Click **Submit** to create a local source for the *Item* data type.
  - b. Save the data type.
11. Switch to the Views tab and create the UI for the new data type.
12. Click **Create new view** name the view ItemDetail with Applies-To of PGPImp-DemoPGP-Data-Entity-Item.
  - a. Place all the fields from the *Item* data type.
  - b. Click **Submit** to save the view.
13. Click **Create new view** name the view ItemDetailRO with Applies-To of PGPImp-DemoPGP-Data-Entity-Item.
  - a. Place all the relevant properties from the *Item* data type, with the *Presentation -> Edit Options* set to *Read-Only*.
  - b. Click **Submit** to save the view.
14. Save the Data type.

## Extending the Work Ancestor Class

Add a Page property to the implementation layer work ancestor of type *PGPImp-DemoPGP-Data-Entity-Item* to hold an *Item* instance during case management processes. The new embedded page will hold (contain) an instance of the new data class embedded on the work.

1. To add a new property, click **Create > Data Model > Property**.
2. In the **Label** field, enter **Item**.
3. In the **Apply to** box, enter, or select PGPImp-DemoPGP-Work.
4. Click **Create and open**.
5. On the **General** tab, in the **Property type** section, click **Change**.
6. Click **Single Page**.
7. In the **Page definition** box, enter or select PGPImp-DemoPGP-Data-Entity-Item.
8. Click **Save**.

- **Entities**
- **New entity overview**

- **Development steps**
- **Rule inventory**
- **Entity data type rules**

## Entities

PGP entities are a collection of robust data structures that model data elements that are commonly used in government processes. The PGP entities are persistent data classes that come pre-configured as a part of PGP.

- Person
- Business
- Vehicle
- Facility
- Household

Entity data structures serve two distinct purposes within PGP: first, Entity instances are persisted to a data store like any local data type or data table; second, Entity data classes are used as a type for embedded pages defined on the PGP work ancestor to support case processing.

## Persistent Data Store

An entity data class (Data- class) is managed (curated) in an independent data store (database table) by the *Add, Update, View and Search* case types (Work- classes).

## Case Type embedded page

An entity data class is also used as a type for embedded pages incorporated into the PGP work ancestor. These embedded pages are used by the PGP component processes to implement PGP features.

PGP Entities follow a pattern of Persistent Data curated by Business Processes. This pattern allows for the possibility that in a customer implementation the entity

datastore may be mapped to customer resource that is external to Pega and PGP. In this pattern a persistent datastore descended from *PegaPS-Data-Entity* is added, updated, viewed, and searched by Case Types descended from *PegaPS-Work-Entity*.

## New entity overview

This guide will step through the process of extending the PGP Entity data structure adding a new entity called Item built in a PGP implementation layer.

In the following examples *PGPImp* is the organization name of the implementation application and *DemoPGP* is the name of the implementation application. The class *PGPImp-DemoPGP-Work* inherits (directed) from the *PegaPS-Work* class.

### Creating the persistent datastore

#### **PGPImp-DemoPGP-Data-Entity-Item**

- Properties relevant to the entity
- View(s) for editing, reviewing
- Data Pages for retrieving
- Optional: Case match rule to avoid duplicates, and validation rules for data

### Extending the work class properties

In the primary work class of the implementation application, *PGPImp-DemoPGP-Work*, add a page property of type *PGPImp-DemoPGP-Data-Entity-Item* to hold *Item* data during case management processing.

### Creating the data management case types

PGPImp-DemoPGP-Work-Entity-Item-Add

PGPImp-DemoPGP-Work-Entity-Item-View

PGPImp-DemoPGP-Work-Entity-Item-Search

## PGPImp-DemoPGP-Work-Entity-Item-Update

### Enabling for App Studio

Use views and design templates to construct the user interface.

Catalog the new Entity's rules in Relevant Records so they will appear in the App Studio smart prompts.

Configure whether case types will appear on +Create nav.

## Development steps

While it is possible to do some of this work in App Studio, the approach discussed here is based on a Dev Studio experience. App Studio hides some development details, like class names and rulesets, to present a development environment that can focus on case types and business processes, but for this exercise we need full control over those details.

### Creating persistent datastore

Create a persistent data type and supporting rules for the new entity.

### Creating data type ancestor

Add an abstract data class to the implementation layer to leverage the Enterprise Class Structure pattern and provide extension points for reusable rules common to the Entity implementation layer data types.

1. To create a new class, click **Create > SysAdmin > Class**.
2. In the **Label** field, enter Entity.
3. In the **Class Name** box, enter or select PGPImp-DemoPGP-Data-Entity.
4. Click Create and open.
5. In the **Select class type** list, select Abstract.
6. In the **Settings** section, in the **Created in version** box, enter or select the current version.

7. In the **Class inheritance** section, in the **Parent class (Directed)** box, enter or select PegaPS-Data-Entity.
8. Click **Save**.

## Creating data type

Create the new *Item* data type and supporting rules.

1. In the navigation pane of Dev Studio, click **Data types**.
2. Click Options > Add data type.
3. In the **Add Data Type** dialog box, click **New Data Type**.
4. In the **Label** field, enter **Item**.
5. In the **Description** field, enter an appropriate description for the class.
6. Expand **Advanced** section.
7. In the **Parent class** box, enter or select PGPImp-DemoPGP-Data-Entity.
8. Click **Submit** to create the new data type and open it for editing
9. On the **Data model** tab, use the Add field button to add properties the *Item* data type
  - a. Add a property named EntityID of type Text.
  - b. Add a property pyID of type Text.
  - c. Continue adding properties as desired to define the Item data type.
  - d. Save the data type.
10. Switch to the **Sources** tab and click **Create a local source**.
  - a. Click **Submit** to create a local source for the *Item* data type.
  - b. Save the data type.
11. Switch to the Views tab and create the UI for the new data type.
12. Click **Create new view** name the view ItemDetail with Applies-To of PGPImp-DemoPGP-Data-Entity-Item.
  - a. Place all the fields from the *Item* data type.
  - b. Click **Submit** to save the view.
13. Click **Create new view** name the view ItemDetailRO with Applies-To of PGPImp-DemoPGP-Data-Entity-Item.

- a. Place all the relevant properties from the Item data type, with the *Presentation -> Edit Options* set to *Read-Only*.
  - b. Click **Submit** to save the view.
14. Save the Data type.

## Extending the Work Ancestor Class

Add a Page property to the implementation layer work ancestor of type *PGPImp-DemoPGP-Data-Entity-Item* to hold an *Item* instance during case management processes. The new embedded page will hold (contain) an instance of the new data class embedded on the work.

1. To add a new property, click **Create > Data Model > Property**.
2. In the **Label** field, enter **Item**.
3. In the **Apply to** box, enter, or select PGPImp-DemoPGP-Work.
4. Click **Create and open**.
5. On the **General** tab, in the **Property type** section, click **Change**.
6. Click **Single Page**.
7. In the **Page definition** box, enter or select PGPImp-DemoPGP-Data-Entity-Item.
8. Click **Save**.

## Creating Case Types

Manage or curate an entity data class (Data- class) in an independent data store (database table) by the *Add, Update, View and Search* case types (Work- classes).

## Creating Abstract Support Work Classes

When a new implementation layer is generated by the New Application Wizard it only creates descendants for the selected concrete classes. Manually add the missing abstract work classes to leverage the Enterprise Class Structure pattern and provide extension points for reusable rules common to the Entity and Item implementation layer work types.

1. To create a new class, click **Create > SysAdmin > Class**.

2. In the **Label** field, enter Entity.
3. In the **Class Name** box, enter or select PGPImp-DemoPGP-Work-Entity.
4. Click **Create and open**.
5. In the **Select class type** list, select Abstract.
6. In the **Settings** section, in the **Created in version** box, enter or select the current version.
7. In the **Class inheritance** section, in the **Parent class (Directed)** box, enter or select PegaPS-Work-Entity.
8. Click **Save**.
9. Click **Create > SysAdmin > Class**.
10. In the **Label** field, enter Entity Item.
11. In the **Class Name** box, enter or select PGPImp-DemoPGP-Work-Entity-Item.
12. Click **Create and open**.
13. In the **Select class type** list, select Abstract.
14. In the **Settings** section, in the **Created in version** box, enter or select the current version.
15. In the **Class inheritance** section, in the **Parent class (Directed)** box, enter or select PGPImp-DemoPGP-Work-Entity.
16. Click **Save**.

## Creating Management Case Types

Manage or curate an entity data class (Data- class) in an independent data store (database table) by the *Add, Update, View and Search* case types (Work- classes).

1. In the navigation pane of Dev Studio, click **Casetypes**.
2. Click **Add a case type**.
3. In the **Name** field, enter a name for the case type.

 **Note:** Choose the case type name carefully as it is used to form the work class name for the case type. After the case type has been created, you may choose to change the pyLabel for the case type rule to something more specific

4. **Optional:** To change the inheritance model for this case type, expand the **Advanced Settings** section, update the **Derives from (Directed)** and **Derives from (Pattern)** inheritance settings, and then provide rulesets and ruleset versions.
5. Click **Submit**.

The lists of ancestor classes to choose from may not show recently added abstract classes unless you have logged out since they were added or otherwise reset the cache.

## Creating the Add Case Type

In PGP the *Add* case type for Entities collects data from the operator in a Page embedded on the work class. At an appropriate step in the case life cycle a PGP Persist component is placed which writes out the data from the embedded page on the work to a persistent data class using the persistence configuration from the *Database Table* rule for the embedded page's class.

1. In the navigation pane of Dev Studio, click **Casetypes**.
2. Click **Add a case type**.
3. In the **Name** field, enter **Add**.



**Note:** You can change the label for the case type rule later

4. Expand the **Advanced Settings** section.
5. In the **Derives from (Directed)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the directed ancestor.
6. In the **Derives from (Pattern)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the patterned ancestor.
7. Click **Submit**.
8. Create a workflow for the case type.
9. On the Workflow tab, click **Life cycle**.

- a. In the default Create stage, click FORM STEP to add a Collect Information step.
  - b. Enter **Capture Item Details** as the name for the step.
  - c. Click **Configure view**.
  - d. In the View Configuration add the Item field as a field group.
  - e. Select ItemDetail from the View dropdown to use it as the View for the Item field group.
10. Add a new *Review* stage.
- a. On the Workflow tab, click **Life cycle**.
  - b. In the default Create stage, click FORM STEP to add a Collect Information step.
  - c. Enter **Review Item Details** as the name for the step.
  - d. Click Configure view.
  - e. Use +STEP to add a Collection Information step under the *Review* stage named *Review Item Details*.
  - f. From the step parameters panel (right) click *Configure the View*.
  - g. In the View Configuration add the Item field as a field group.
  - h. Select ItemDetailRO from the View dropdown to use it as the View for the Item field group.
  - i. Use +STEP -> ...More -> Processes and select Persist object.
  - j. From the step parameters panel (right) configure the Persist object component.
    1. For Type use Entity
    2. For Property use "item" (double quotes are significant)
  - k. Change the case type rule label for PGPImp-DemoPGP-Work-Entity-Item-Add from Add to Add item.
  - l. Save the Case.

## Creating the Update Case Type

In PGP the *Update* case type for Entities solicits changes to the entity data from the operator and captures these changes in a Page or Page List embedded on the work class. At an appropriate step in the case life cycle a PGP Persist component is placed which writes out the data from the embedded page on the work to a

persistent data class using the persistence configuration from the *Database Table* rule for the embedded page's class.

1. In the navigation pane of Dev Studio, click **Casetypes**.
2. Click **Add a case type**.
3. In the **Name** field, enter **Update**.



**Note:** You can change the label for the case type rule later.

4. Expand the **Advanced Settings** section.
5. In the **Derives from (Directed)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the directed ancestor.
6. In the **Derives from (Pattern)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the patterned ancestor.
7. Click **Submit**.
8. Create a Workflow for the case type.
  - a. On the Workflow tab, click **Life cycle**.
  - b. In the default Create stage, click FORM STEP to add a Collect Information step.
  - c. Enter **Edit Item Details** as the name for the step.
  - d. Click **Configure view**.
    1. In the View Configuration add the *Item* field as a field group.
    2. Select *ItemDetail* from the View dropdown to use it as the View for the *Item* field group.
    3. Click **Submit** to save the view.
9. Create a new *Review* stage.
  - a. Click **+STAGE** to add a stage and enter Review.
  - b. Click **+STEP > Collect information** and name it *Review Item Details*.
  - c. From the step parameters panel (right) click *Configure the View*
  - d. In the View Configuration add the *Item* field as a field group
  - e. Select *ItemDetail/RO* from the View dropdown to use it as the View for the *Item* field group.

- f. Click **Submit** to save the view.
  - g. Click **+STEP > More > Processes > Persist Object**.
  - h. From the step parameters panel (right) configure the *Persist object* component.
  - i. For Type use Entity
  - j. For Property use “.Item” (double quotes are significant)
10. Change the case type rule label for PGPImp-DemoPGP-Work-Entity-Item-Update from Update to Update item.
11. Save the case.
12. Edit the *pyDefault* data transform rule for the *Update* work type to set the initial values on the *.Item* page from lookup data page.
- a. Switch to the *Parameters* tab and add a parameter named *EntityID* of type *String*
  - b. Add a *Set* action to the transform definition.
  - c. Use *.Item* as the *Target*.
  - d. Use *D\_Item[pyID:Param.EntityID]* as the *Source*.
13. Save the data transform.

## Creating the View Case Type

In PGP the *View* case type for Entities shows a comprehensive snapshot of an entity instance's data.

1. In the navigation pane of Dev Studio, click **Casetypes**.
  2. Click **Add a case type**.
  3. In the **Name** field, enter **View**.
- i **Note:** You can change the label for the case type rule later.
4. Expand the **Advanced Settings** section.
  5. In the **Derives from (Directed)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the directed ancestor.

6. In the **Derives from (Pattern)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the patterned ancestor.
7. Click **Submit**.
8. Create a Workflow for the case type.
  - a. In the name field of the Create stage, enter or select PegaPS-Work-Entity view from the list of available stage processes.
  - b. Change the case type rule label for PGPImp-DemoPGP-Work-Entity-Item-View from View to View item.
9. Click **Save**.
10. Save the PegaPS.Details section as PGPImp-DemoPGP-Work-Entity-Item-View.Details.
  - a. Include the section *PGPImp-DemoPGP-Data-Entity-Item.Details* using the *Item* property as the page context.
11. Save the section.
12. Edit the *pyDefault* data transform rule for the *View* work type to set the initial values on the *.Item* page from lookup data page.
  - a. Switch to the Parameters tab and add a parameter named EntityID of type String.
  - b. Add a Set action to the transform definition.
  - c. Use Item as the Target.
  - d. Use D\_Item[pyID:Param.EntityID] as the Source.
13. Save the data transform.

## Creating the Search Case Type

In PGP the *Search* case type collects filter criteria from the user and retrieves a list of matching entities which may then be selected for further processing.

1. In the navigation pane of Dev Studio, click **Casetypes**.
2. Click **Add a case type**.
3. In the **Name** field, enter **Search**.



**Note:** You can change the label for the case type rule later.

4. Expand the **Advanced Settings** section.
5. In the **Derives from (Directed)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the directed ancestor.
6. In the **Derives from (Pattern)** box, enter or select PGPImp-DemoPGP-Work-Entity-Item as the patterned ancestor.
7. Click **Submit**.
8. In the Stage Name field for the Create stage use the smart-prompt to select the PegaPS-Work.Search stage from the list of available stages.
9. Override/Save As the PegaPS-Work.Search section as PGPImp-DemoPGP-Work-Entity-Item-Search.Search.
10. Edit the PGPImp-DemoPGP-Work-Entity-Item-Search.Search section.
  - a. For Page context choose Use clipboard page.
  - b. For Class enter PGPImp-DemoPGP-Data-Entity-Item.
  - c. For Clipboard page select Item from the list.
  - d. For Section choose by name and select Search from the list.
  - e. For PARAMETER ShowSearchType set Value to nothing.
  - f. Click **Submit**.
12. Save the section.
13. Override/SaveAs the PegaPS-Data.SearchCriteria section as PGPImp-DemoPGP-Data-Entity-Item.SearchCriteria.
  - a. Add the key properties for PGPImp-DemoPGP-Data-Entity-Item to this section.
  - b. Save the section.
14. Override/SaveAs the PegaPS-Data.SearchResults section as PGPImp-DemoPGP-Data-Entity-Item.SearchResults.
15. Edit the PGPImp-DemoPGP-Data-Entity-Item.SearchResults section.
16. Use Structural > Table to add a table layout to the section and Configure the table.
  - a. For Source choose Data page.
  - b. For Data page select D\_ItemList from the list.
  - c. For Parameters check Pass the parameter page.

- d. Submit.
17. Save the section.

## Enabling for App Studio

Socialize the new rules with the dev environment so that they work with, for example, smart-prompting.

- Use views and design templates to construct the user interface.
- Manage the new Entity's rules in Relevant Records so that appropriate choices will appear in the App Studio smart prompts.
- For appropriate case types set the *Show in 'New' menu* checkbox under Settings -> General to enable the case type to appear on +Create nav.

## Rule inventory

The following is a list of rules created in this exercise.

Applies -to	Rule name	Rule type	Description	Comment
	PGPImp-DemoPGP-Data-Entity	Class	Abstract	
	PGPImp-DemoPGP-Data-Entity-Item	Class	Concrete	Key EntityID
	PGPImp-DemoPGP-Data-Entity-Item	Database Table	Persistence	
PGPImp-DemoPGP-	ItemName	Property	Text	

Applies -to	Rule name	Rule type	Description	Comment
Data-Entity-Item				
PGPImp-DemoPGP-Data-Entity-Item	ItemType	Property	Text	
PGPImp-DemoPGP-Data-Entity-Item	ItemDetail	Section	View	
PGPImp-DemoPGP-Data-Entity-Item	ItemDetailRO	Section	View	
	D_Item	Data Page	Page	Source=Lookups
	D_ItemList	Data Page	Page List	Source=Report Definition
PGPImp-DemoPGP-Data-Entity-Item	ItemList	Report Definition		All Item Properties
PGPImp-DemoPGP-Work	.Item (PGPImp-DemoPGP-Data-Entity-Item)	Page		
PGPImp-DemoPGP-Work	.ItemList (PGPImp-DemoPGP-	Page		

<b>Applies -to</b>	<b>Rule name</b>	<b>Rule type</b>	<b>Description</b>	<b>Comment</b>
	<i>Data-Entity-Item)</i>			
	PGPImp-DemoPGP-Work-Entity	Class	Abstract	
	PGPImp-DemoPGP-Work-Entity-Item	Class	Abstract	
	Add	Case Type		
	PGPImp-DemoPGP-Work-Entity-Item-Add	Class	Concrete	Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	pyDefault	Case Type		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	CreateForm_Default	Flow		Case Type
	D_Add	Data Page		Case Type
	D_AddList	Data Page		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	pyDefault	Data Transform		Case Type

Applies -to	Rule name	Rule type	Description	Comment
PGPImp-DemoPGP-Work-Entity-Item-Add	pySetFieldDefaults	Data Transform		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	ItemInformation_Flow	Flow		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	Create	Flow Action		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	Create	Section		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	ItemDetails	Flow Action		Case Type
PGPImp-DemoPGP-Work-Entity-Item-Add	ItemDetails	Section		Case Type

## Entity data type rules

Switch to App Explorer (App) and show PGPImp-DemoPGP-Data-Entity-Item. Review all the rules created for the Item data entity.

Applies to	Rule name	Rule type	Description	Comment
	PGPImp-DemoPGP-Data-Entity	Class	Abstract	
	Item		Data Type	
	PGPImp-DemoPGP-Data-Entity-Item	Class	Concrete	Key pyID
	PGPImp-DemoPGP-Data-Entity-Item	Database Table	Persistence	
PGPImp-DemoPGP-Data-Entity-Item	ItemName	Property	Text	
PGPImp-DemoPGP-Data-Entity-Item	... more properties	Property	Text	
	D_Item	Data Page	Page	Source=Lookup
	D_ItemList	Data Page	Page List	Source=Report Definition
	D_ItemSavable	Data Page	Page	
PGPImp-DemoPGP-	DataTableEditorReport	Report Definition		All Item Properties

Applies to	Rule name	Rule type	Description	Comment
Data-Entity-Item				

## Appendix C: Mobile Strategy

### Intended audience

This guide is intended for Pega Government Platform (PGP) developers designing mobile and mobile offline applications in Pega 8.8 using Pega Government Platform 8.8.

### Guide organization

This guide contains the following sections:

Pega Mobile overview	Provides an overview of Pega Mobile.
PGP Mobile	Describes how PGP has been configured to support Pega Mobile applications.
PGP Mobile Offline	Describes how PGP has been configured to support Pega Mobile applications.

- [Pega Mobile overview](#)
- [Pega Government Platform Mobile](#)
- [Mobile Offline](#)
- [Changes for Mobile Offline](#)
- [Changes to Pega Government Platform features](#)
- [Pega JavaScript API](#)

# Pega Mobile overview

## Mobile solutions

Mobile solutions provide government developers with several features and scenarios that help develop and deploy mobile apps to suit different businesses. Professionals and citizen developers can work in the model and take advantage of these features to create a modern mobile experience.

With low-code authoring, you can create a mobile channel for your application and effectively configure the content of the app in the model. For more information, see [Mobile solutions](#).

## Pega 8 Mobile overview

Pega provides four mobile deployment models: Device Browser, Pega Mobile Preview (previously Pega Mobile Express), Pega Mobile Deployment Kit, and Custom Mobile App.

### Device Browser

Pega web applications built with responsive UI can be accessed from any device, for example, any tablet or any smartphone, using the device's native web browser, such as Google Chrome, Apple Safari, Microsoft Edge, and so on. No app store or developer licensing accounts are needed to follow this model.

### Pega Mobile review

For testing purposes, Pega provides the Pega Mobile Preview app available as a free registered download from the Google Play Store for Android, or from the App Store for iOS (See also Pega Mobile Express). The Pega Mobile Preview app is a generalized Pega application launcher that takes the place of a web browser providing application access to device services like GPS and Camera with configurable security controls that a browser-based interaction does not have.

## Pega Mobile Deployment kit

After the testing is complete, Pega can generate an application-specific mobile deployment kit much like Pega Mobile Preview, but custom tailored to the specific application channel and target platform. This allows the customer to have complete control over kit distribution.

## Custom Mobile App with Pega Connect SDK

Experienced mobile application authors can build completely custom mobile apps and enable them to interact with Pega via the Pega Mobile SDK. For more information, see [Mobile features overview](#).

## Mobile Offline

Offline support ensures that key work scenarios are available to your users when they work in the field or in situations with limited or no connectivity.

The following scenarios are available as offline support for your application

### Field workers support

Maintain business continuity and assist users with specific offline tasks.

### Data collection

Collect data in the form of a questionnaire.

### Data always available

Provide field workers with offline data catalogs that are necessary to perform their tasks, such as offline catalogs of addresses or parts.

### Case attachments

Use attachments in cases, such as videos or PDFs that are seamlessly available in offline mode. Mobile users can also add new attachments to cases and ensure that these attachments are available to other users when the app synchronizes with the server.

# Pega Government Platform Mobile

## Business Issue

Government customer requirements for applications include deployment to mobile devices like tablets and cell phones.

## PGP solution

In PGP 8.x all components have been enabled for mobile operation and feature a responsive UI that adjusts automatically to different screen sizes and input devices.

## Customer use

Application case types built in App Studio automatically benefit from the mobile-aware responsive UI.

# Mobile Offline

## Business issue

Government customer requirements for mobile applications include operation in conditions where mobile devices move in and out of areas with no cellular or wireless reception.

## PGP Solution

In PGP, the following core components have been enabled for mobile offline operation.

- Capture interview
- Conduct assessment (simple survey questions only)
- Activity plan task
- Add Evidence

Mobile applications built with these components will automatically store data when offline conditions are encountered and synchronize the stored data with server when offline condition is resolved.

More PGP components are enabled for offline operation in future releases.

## Customer use

Customers who wish to use the offline capabilities in PGP implementations must follow three steps.

1. Enable *Offline* support in the Mobile Channel Settings.
2. Create and populate an *Offline Configuration* rule.
3. Develop mobile application case types in App Studio using one or more of the PGP Mobile Offline enabled components.

### Enabling offline support

Enable offline support to advance cases even if the users' devices are not connected with the server.

1. Navigate to the mobile channel definition for your application.
2. Click the Configuration tab, and then click the Offline category.
3. Select the Enable offline support check box.
4. Click the Add case type link and select the application case types enabled for use in the offline mode.
5. Click Submit, and then click Save.

For more information, see [Setting up mobile apps](#).

### Creating an offline configuration rule

Manage data synchronization as a device transitions between online and offline modes and identify data elements for Pega to explicitly manage. The Offline Configuration rule (Rule-Config-Offline) allows a developer to list application data elements that are managed offline.

To create or edit an Offline Configuration rule, navigate to Configure > Mobile > Offline Configuration.

In Pega, you can manage the following data resources offline.

- Data Pages
- Field Values
- Data Transforms
- Validate rules
- When rules
- Decision rules
- Decision Tree rules

For an example of an Offline Configuration rule, review the one provided for ICM: PGPICM.

For more information, see [Working with offline-enabled mobile apps](#).

### **Develop offline case types**

Customers can design and develop Pega Mobile Offline case types following the guidance provided by the Pega Community.

## **Changes for Mobile Offline**

### **New offline configuration rule**

PGP provides a PGP-specialized Offline Configuration rule template for ICM that manages mobile offline data synchronization on the ICM Mobile Channel for the channel, its portal, and Interview, Conduct Assessment, and Activity Plan Task components.

### **Allow list - Data pages**

D\_AppContext  
D\_ICMAppContext  
D\_OptionMapListAll  
D\_InterviewTemplateListAll  
D\_StatesListAll  
D\_ConfiguredCountryList  
D\_InvestigationListAll  
D\_SubjectListAll

D\_AuthorizedInvestigations  
D\_Association\_Editable

### **Field values**

no field values

### **Data transforms**

PegaPS-Work!InitializeInterview  
PegaPS-Data-Interview!LoadInterviewNotes  
PegaPS-Data-Address!SetStatusLabel  
PegaPS-Data-Interview!SetInterviewTitle  
PegaPS-Work!ReassignAssessment  
PegaPS-Work!PostAddInterview\_Ext

### **Validation rules**

no validate rules

### **when rules**

Work-!pyCanDisplayAttachments  
Work-!CanDisplayDocuments  
Work-!pyHasAttachments  
@baseclass!Never  
Assign-!pzHasPrivilegeToClearAssignmentErrors  
@baseclass!pzIsOfflineEnabled  
PegaPS-Work!IsNotNewAndResolved

### **Decision Table rules**

PegaPS-Data-Address!StatusLabel

### **Decision Tree rules**

no decision tree rules

### **Deny list - Excluded data pages**

D\_UITemplateMeta

D\_pzClientStore\_Item  
D\_pzClientStore\_Action  
D\_pzClientStore\_Category  
Declare\_pyDisplay  
D\_pzOfflineAttachmentList  
D\_pzManageExpress

### **Delta-Sync Deny list - Excluded data pages**

D\_QuestionOptions\_Master  
D\_QuestionOptions  
D\_StartingFlows  
D\_pzOfflineCaseAttachments  
D\_pzSearchConfiguration  
D\_pzOfflineSettings

### **Resources images and icons**

webwb/zblankimage.gif  
images/busyIndicator.gif  
images/pyEmailThumbnail.svg  
images/pyURLThumbnail.svg  
images/pyTextThumbnail.svg  
images/pyPDFThumbnail.svg  
images/pyWordThumbnail.svg  
images/pyPPTThumbnail.svg  
images/pyExcelThumbnail.svg  
images/pyImageThumbnail.svg  
pyDefaultFileThumbnail.svg  
webwb/webwb/caret-down.svg  
webwb/webwb/pycheckboxes.png  
webwb/webwb/calendar-time-date.svg  
webwb/pymodal\_mask\_img.png  
webwb/requiredstar.gif  
webwb/pzerror\_icon.png

webwb/py-calendar-icon.svg  
webwb/pymodal\_mask\_img.png  
webwb/requiredstar.gif  
webwb/pzerror\_icon.png  
webwb/webwb/caret-down.svg  
webwb/py-calendar-icon  
webwb/webwb/pycheckboxes.png  
webwb/webwb/calendar-time-date.svg  
webwb/pyConnected.svg  
webwb/zblankimage.gif  
images/busyIndicator.gif  
images/pyEmailThumbnail.svg  
images/pyURLThumbnail.svg  
images/pyTextThumbnail.svg  
images/pyPDFThumbnail.svg  
images/pyWordThumbnail.svg  
images/pyPPTThumbnail.svg  
images/pyExcelThumbnail.svg  
images/pyImageThumbnail.svg  
images/pyDefaultFileThumbnail.svg  
images/pzPegalcon.ico  
images/pzPegalcon16.png  
images/pzPegalcon32.png

## Fonts

webwb/webwb/pxfont-OpenSans-Bold.woff2  
webwb/webwb/pxfont-OpenSans-BoldItalic.woff2  
webwb/webwb/pxfont-OpenSans-ExtraBold.woff2  
webwb/webwb/pxfont-OpenSans-ExtraBoldItalic.woff2  
webwb/webwb/pxfont-OpenSans-Italic.woff2  
webwb/webwb/pxfont-OpenSans-Light.woff2  
webwb/webwb/pxfont-OpenSans-LightItalic.woff2  
webwb/webwb/pxfont-OpenSans-Regular.woff2

webwb/webwb/pxfont-OpenSans-Semibold.woff2  
webwb/webwb/pxfont-OpenSans-SemiboldItalic.woff2  
webwb/pxfont-OpenSans-Bold.woff2  
webwb/pxfont-OpenSans-BoldItalic.woff2  
webwb/pxfont-OpenSans-ExtraBold.woff2  
webwb/pxfont-OpenSans-ExtraBoldItalic.woff2  
webwb/pxfont-OpenSans-Italic.woff2  
webwb/pxfont-OpenSans-Light.woff2  
webwb/pxfont-OpenSans-LightItalic.woff2  
webwb/pxfont-OpenSans-Regular.woff2  
webwb/pxfont-OpenSans-Semibold.woff2  
webwb/pxfont-OpenSans-SemiboldItalic.woff2  
webwb/uikit-icons.woff2

## **Locales Locale ID**

no locales

For more information, see [Working with offline-enabled mobile apps](#).

## **PGP Offline JavaScript support library**

The *pyPega\_UI\_UserScripts\_Offline* Static Content Bundle rule has been extended to include the *pega\_pgp\_utils.js* Text File rule, a PGP JavaScript library that facilitates some specialized PGP data management for offline scenarios.

The *pega\_pgp\_utils.js* JavaScript library contains functions that help manage data in the mobile device's data cache while in an offline state using the Pega *Client Cache API*.

## **Changes to Pega Government Platform features**

### **Interview Component**

The *InterviewInfo* Section rule was modified to call several of the JavaScript functions in *pega\_pgp\_utils.js*. In the Change Action for the Interview Template drop-down list, a Run Script was added to the Action Set to call the *LoadInterviewNotes()* function to retrieve

the selected template from the mobile device's client cache and populate the *InterviewNotes* property when in an offline state.

Similarly for the *NonMandatoryWithCurrentLocation* Section rule, the Country dropdown calls the *setStateLabel()* function to populate the states for a selected country.

The *PersistInterview* Declare Trigger rule has been added to save an Interview instance during the data synchronization.

The *MoveCaseAttachments* Declare Trigger rule has been added to associate attachments collected while in an offline state with the Interview data instance.

## Conduct Assessment Component

For ICM Assessment (Pega Survey), PGP overrides a Pega Survey extension Work-.pyResolveSurvey Flow rule as PegaSP-Work.pyResolveSurvey and extends it to associate an Assessment to an Investigation or Subject.

## Activity Plan Task case type

The *TaskInformation* Section rule has been modified to include the Pega Work-.pyOfflineCaseAttachments Section rule which conditionally displays in the place of the *PegaPS.ViewAttachments* Section rule when *pzIsOfflineEnabled* is true.

## Pega JavaScript API

JavaScript APIs for offline-enabled mobile apps

<https://docs-previous.pega.com/mobile/87/javascript-apis-offline-enabled-mobile-apps>

Mobile Offline Data-Sync Architecture

[https://agilestudio.pega.com/prweb/AgileStudio/app/agilestudio\\_/doc/DOC-6234](https://agilestudio.pega.com/prweb/AgileStudio/app/agilestudio_/doc/DOC-6234)

# Limit the Visibility of sensitive data items through attribute-based access control (ABAC)

## What is Attribute based access control?

Attribute-based access control (ABAC) and role-based access control (RBAC) are two ways of controlling the authentication process and authorizing users.

The difference between ABAC and RBAC is that ABAC provides access rights based on user, environment, or resource attributes, for example, a data page on the clipboard, while RBAC provides access to resources or information based on user roles, such as an Access Role or Access Role to Object rule. Essentially, RBAC controls broad access across an organization, while ABAC takes a fine-grain approach.

Pega Platform provides rules that you can use to implement access control by using values of attributes that are present on the clipboard:

- Access Control Policy
- Access Control Policy Condition
- Access When

## Use case examples

Use the ABAC security feature in Pega Platform to mask sensitive data, such as personally identifiable information (PII), so that any unauthorized users cannot see it.

In the following examples, the ABAC Access Control Policy rule is used to mask sensitive data that is stored in `Tax ID`, `Security question` and `Security answer` properties of an instance of a `Person` data entity so that an unauthorized user cannot see the property values:

- A non-administrative end user can add, view, and update the `Person` entity instances that they own, but they cannot view sensitive data of instances they do not own.
- An administrative end user, such as an operator with the manager access role, can add, view, and update all the sensitive data of any `Person` entity instances.
- A general case worker can add a new `Person` entity instances and enter the `Tax ID`, `Security question`, and `Security answer` values, but they cannot view or update the sensitive data properties after adding the entity.

**Update Person (UP-8016) OPEN**

<p>Country *</p> <input type="text" value="US"/>	<p>State *</p> <input type="text" value="Virginia"/>		
<p>City *</p> <input type="text" value="Busdrive"/>	<p>Postal code *</p> <input type="text" value="48075"/>		
<p><b>Communication details</b></p> <p>Home phone</p> <input type="text" value="(248) 286-2505"/>		<p><b>Communication preferences</b></p> <p>Method</p> <input type="text" value="Select"/>	
<p>Mobile phone</p> <input type="text"/>		<p>Time</p> <input type="text" value="Select"/>	
<p>Email *</p> <input type="text" value="Alekhya.Pudu@in.pega.com"/>		<p>Language</p> <input type="text" value="Select"/>	
<p><b>Verification details</b></p> <p>Tax ID *****9879</p> <p>Security question 1 * *****</p> <p>Security question 2 * *****</p> <p>Security answer * *****</p> <p>Security answer * *****</p>			
<input type="button" value="Cancel"/> <input type="button" value="Save"/> <input type="button" value="Submit"/>			

**Form** - A view of your data that displays information or collects input from users as they create, update, and resolve cases in your application. Open your case type from the Application Explorer to access the options for configuring a form.

Person ID	Last name/Surname	First name	Date of birth	Tax ID	Association details
PERSON-895	LName893	FName8938	Sep 4, 2019	*****5525	<a href="#">View</a>
PERSON-894	edyery	dfhf	Sep 4, 2019	*****4356	<a href="#">View</a>
PERSON-893	LName893	FName8933	Sep 3, 2019	*****7545	<a href="#">View</a>
PERSON-876	LastName-1096	FirstName-1096	Jun 16, 1982	*****8565	<a href="#">View</a>
PERSON-826	LastName-1011	FirstName-1011	Jun 16, 1982	*****8565	<a href="#">View</a>
PERSON-781	LastName-933	FirstName-933	Jun 16, 1982	*****8565	<a href="#">View</a>
PERSON-736	LastName-853	FirstName-853	Jun 16, 1982	*****8565	<a href="#">View</a>
PERSON-710	LName893	FName893	Sep 4, 2019	*****8568	<a href="#">View</a>

*Search person*

*Update Person*

## Implement attribute-based access control

The following content represents the ways to implement the attribute-based access control.

To implement this example, two Access Control Policy rules were created:

- PropRead .TaxID
- PropRead .QuestionAndAnswers

To allow the grant of custom permissions, two Access Control Policy Condition rules were created

- ViewTaxID
- ViewQuestionAndAnswers

**Access Control Policy: Tax ID [Available]**  
CL: PegaPS-Data-Identifier-SSN ▾ ID: PropRead • TaxID RS: PegaPS:08-04-01

**Definition** Specifications History

**TARGET**  
PegaPS-Data-Identifier-SSN

**ACTION**  
PropertyRead

**APPLIES TO**  
Listed properties only

Disallow creation of a policy with the same name at a descendant class

Permit access if  is true.

Property	Restriction Method
.TaxID	<input type="button" value="Mask all but last 'N'"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
.IDNumber	<input type="button" value="Mask all but last 'N'"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

[+ Add property](#)

*Access Control Policy: PropRead TaxID*

**Access Control Policy Condition: View tax ID [Available]**  
**CL:** PegaPS-Data-Identifier-SSN **ID:** ViewTaxID **RS:** PegaPS:08-05-01

Save as Actions Private edit X

Definition Pages & Classes Specifications History

---

### Conditional Logic

- WHEN  
WhenSearchPerson Permit access if B
- WHEN  
IsSearchPersonTable Permit access if B
- WHEN  
WhenMerge Permit access if C
- WHEN  
IsManagerOrAdmin Permit access if B OR D
- OTHERWISE  
Permit access if B OR (A AND D)

[+ Add conditional logic](#)

---

### Policy Conditions

Condition	Column source	Relationship	Value	
A	.EntityID	Is equal	D_UserProfile.EntityID	<input type="button" value="Delete"/>
B	.EntityID	Is null		<input checked="" type="checkbox"/> Treat Empty As Null <input type="button" value="Delete"/>
D	.ViewIDNumber	Is equal	true	<input type="button" value="Delete"/>
C	.EntityID	Is not null		<input checked="" type="checkbox"/> Treat Empty As Null <input type="button" value="Delete"/>

[+ Add Condition](#)

### Access Control Policy Condition: ViewTaxID

**Access Control Policy: Question and answers [Available]**  
**CL:** PegaPS-Data-Question **ID:** PropRead • QuestionAndAnswers **RS:** PegaPS:08-04-01

**Definition** Specifications History

**TARGET**  
PegaPS-Data-Question

**ACTION**  
PropertyRead

**APPLIES TO**  
Listed properties only

Disallow creation of a policy with the same name at a descendant class

Permit access if  ViewQuestionAndAnswe is true.

Property	Restriction Method
.Answer	Full Mask
.Question	Full Mask

[+ Add property](#)

---

**Access Control Policy Condition: View question and answers [Available]**  
**CL:** PegaPS-Data-Question **ID:** ViewQuestionAndAnswers **RS:** PegaPS:08-04-01

**Definition** Pages & Classes Specifications History

**Conditional Logic**

- WHEN**  WhenForgotPassword Permit access if  C
- WHEN**  IsManagerOrAdmin Permit access if  B OR D
- OTHERWISE**  Permit access if  B OR (A AND D)

[+ Add conditional logic](#)

**Policy Conditions**

Condition	Column source	Relationship	Value
A	.EntityID	Is equal	<input type="text"/> D_UserProfile.EntityID
B	.EntityID	Is null	<input checked="" type="checkbox"/> Treat Empty As Null
D	.ViewData	Is equal	<input type="text"/> true
C	.pxObjClass	Is not null	<input checked="" type="checkbox"/> Treat Empty As Null

[+ Add Condition](#)

*Access Control Policy: ViewQuestionAndAnswers*

*Access Control Policy: PropRead QuestionAndAnswers*

# Leverage Pega Government Platform from Pega Customer Service

- Pega Government Platform and Pega Customer Service Implementation

## Pega Government Platform and Pega Customer Service Implementation

- Business Objective
- Purpose
- Solution
- Implementation steps

### Business Objective

In the past, Pega Government Platform customers have found it challenging to integrate PGP with Pega Customer Service in a manner that enables the best of both products' worlds: government entity management and communications channel management. Working through the initial integration of these two products has been known to take two or more weeks out of a development schedule. For sales demonstration teams and customers, this represents a significant hardship.

### Purpose

PGP product management provides an integration approach and a supporting ruleset to greatly reduce the PGP + CS project setup time so that our customers can focus on the business of specifying case types and not be distracted by initial product stack configuration.

### Solution

Several plausible solution architectures are developed in response to these challenges (see ###whitepaperxxxx###, and ###whitepaperyyyy###). The PGP team have chosen

to provide a CS implementation layer ruleset which contains rule overrides for common extensions from PGP and CS enabling the high-value features from both products.

- All PGP core case types are enabled for use in CS, with full interaction driver support
- Person View and Business View integrated
- CS contact search integrated to PGP person entity
- CS business search integrated to PGP business entity
- Shared work pool: Cases created in the CS layer can be processed in the PGP layer
- Pega Call Demo-Pop support for common PGP Demo personas. For example, Marty Solomon

By leveraging the PGP + CS Bridge (GovCSIntegration85.zip), this effort to build the CS+PGP initial integrated product stack can be greatly reduced, and on-premises customers can assemble the application stack and be ready to build an application on PGP and CS under 2 hours.

This integration kit is available on the Pega Marketplace at the following URL

<https://community.pega.com/marketplace/components/pgp-customer-service-integration-ruleset>

## Implementation steps

1. If you have existing implementation layer application built on Pega Government Platform v8.5, then you can go to step 3.
2. To build a new implementation layer on top of Pega government platform application then follow the PGP implementation guide, <https://community.pega.com/knowledgebase/documents/pega-government-platform-85-implementation-guide>
3. For instance, if you have built PGP Demo application on Pega Government Platform (or ICM, Procurements, Complaints) application, then open the application definition and add CustomerService application to the built-on

applications stack and GovCSIntegration ruleset to Application ruleset stack as shown below.

4. Logoff and login again to reset the ruleset stack.
5. For case types that were already built in the implementation layer (i.e, only for old case types), go to records explorer, under Process select intent task, click on create and give apply to class as “PegaCA-Work-Interaction” to show intent task under General category.
6. Note: For new case types created, the Intent task will be created automatically.
7. Mention the case type’s class in Task class field and starting flow as below and save it.
8. Creation of access groups
  - a. Search for CSCSR access group and save as with a new name and mention your application in the application field as shown below.
  - b. Creation of new operator with created access group:
  - c. Go to records explorer, under organization select Operator ID. Click on Create.
  - d. Add above created access group to the operator as shown below and under work tab have team as “CSR” and save.
9. Repeat steps 7,8 for other access groups CSBOManager, CSBOUser, CSManager if required.
10. For new case types that are created after updating the application record, intent tasks get created automatically under Account category. To change the category under which the intent task to be displayed, from Case types explorer, open the case type, click on settings tab, Go to Service request options and change the category.
11. To enable the use of App studio, add CS:ExpressMgrTools, CS:ExpressUser roles to the admin access group.

# Integrating Industry Applications with Pega Customer Service

The Pega Customer Service (CS) product offers features needed to build typical customer service representative (CSR) call-center support applications. CS provides a CSR/Operator conversational dialogs woven together with service tasks (case types) presented through an intent-driven user interface called the Interaction Driver.

The screenshot shows the Pega Interaction Driver interface. On the left, there's a sidebar with navigation links like '+Add Task', 'Manage relations', 'Manage relations Wra...', 'DemoCase', and 'Wear Up'. The main area has four sections: 'CONTACT INFORMATION' (Callback: 617-839-5764, Phone: (617) 454-6356, Email: sarah.smith.csp@gmail.com, Address: 120 Avon Hill Street, Cambridge, MA, 02130), 'CUSTOMER SUMMARY' (Open cases: 0, Active accounts: —, Communication preference: Phone), and 'RELATIONSHIP' (Churn risk: —, NPS trend: Passive, Lifetime value: —, Customer since: —). A red box highlights a message in a conversation window: 'Hello \*unknown\*, may I have few details regarding the complaint that you have?'. Below the main sections is a form for 'Enter Person Details (AP-33)' with fields for Salutation (Select), Last name / Surname (Required), First name (Required), Middle name (Select), Suffix (Select), Gender (Male/Female), and a 'SHOW LESS' button.

With no additional development a CS process can invoke any case type available in the application rule stack. Given a specific case type, CS will:

- Call the rule-resolved pyStartDefault activity for the specified case type's work class
- Pass process control to the resulting flow
- Receive process control when that flow ends

If the case type invoked in this way is a descendant of main CS case type ancestor class PegaCA-Work, then the flow UI will present the Interaction Driver, which is embedded in the default CS harnesses, along with any dialogs and tasks associated with the case type. However, if the case type does not have an ancestor of PegaCA-Work, For example, is from an underlying Built-On application, then some development effort will be required to extend the benefits of the Interaction Driver to those underlying Built-On case types.

The CS Interaction Driver is implemented as a customization to standard Pega harnesses. The key components are:

1. The PegaCA-Work sections CPMPPerformIncludes and CPMPPerformIncludesTasks which implement the Interaction Driver User Interface. These sections are included in the following CS harnesses:
    - New
    - Perform, PerformScreenFlow
    - TabbedScreenFlow, TabbedScreenFlow7
    - TreeNavigation7
  2. The PegaCA-Work activity NewDefaults which manages the initialization of the Interaction Driver and Dialog data structures.
  3. The PegaCA-Work activity PerformDefaults which manages the list of case types in the Interaction Driver's task area.
  4. CS Service Tasks.
  5. CS Scripted Dialogs associated with steps in a case flow.
- [\*\*Case Study: Extending Pega Government Platform with Pega Customer Service\*\*](#)

## Case Study: Extending Pega Government Platform with Pega Customer Service

The following tutorial will create a new Customer Service implementation stack that extends the PGP Complaints application fully supporting the CS Interaction Driver in the underlying Built-On application's case types.

- [\*\*Overview\*\*](#)

## Overview

1. Create a new CS Implementation layer using the New Application wizard.
2. Edit the new application to include PGPComplaints in the list of Built-On applications.

3. Override the Perform and Confirm harnesses for the PGPComplaints case types in the new implementation layer.
4. Override the NewDefaults and PerformDefaults activities for the PGPComplaints case types in the new implementation layer.
5. Override the DPLoadContact activity for the CS Contact list in the new implementation layer and modify it to populate the CS AppContact page with the values from the PGP Complaints D\_Person data page (Pass the Parameter ContactId to Data page param ID).
6. Override the LoadOptionMapList data transform in the new implementation layer and modify it to load option pairs for PGP drop-down lists from the appropriate implementation layer data page.
7. Override the PegaPS-Data-Context-Application D\_AppContext data page (dynamic class referencing) in the new implementation layer with the D\_AppContext from the new CS implementation layer.
8. Add CS appropriate CS Dialogs and associate them with PGPComplaint case types.

## Driving UI with Local Data

- Option Map
- Architecture
- Using OptionMap Data in a Picker control

### Option Map

The PGP Option Map is a pattern for supplying customizable, configurable data collections to Pega *Picker* user interface selection controls like *pxAutoComplete* and *pxDropdown* (drop-down list).

Name of college or university\*

*Education Autocomplete*

The following are the benefits of this pattern:

- Instances can be managed from App Studio using the Data Type Record Editor in support of the low-code/no-code developer.
- Value display order can be explicitly controlled (not just alphabetical).
- The solution scales from a few instances to thousands of instances.
- The solution is localizable.
- The data instances are supplied by a parameterized data page can be used as a source in many data driven use cases.
- The data page is loaded by a data transform that may be customized in an implementation layer.

## Architecture

The Option Map consists of three basic elements.

- A Pega local data type that holds the collections of name-value pairs grouped by field name group.
- An interface to manage the *OptionMap* data instances.
- A data page to retrieve collections of name-value pairs based on parameters.

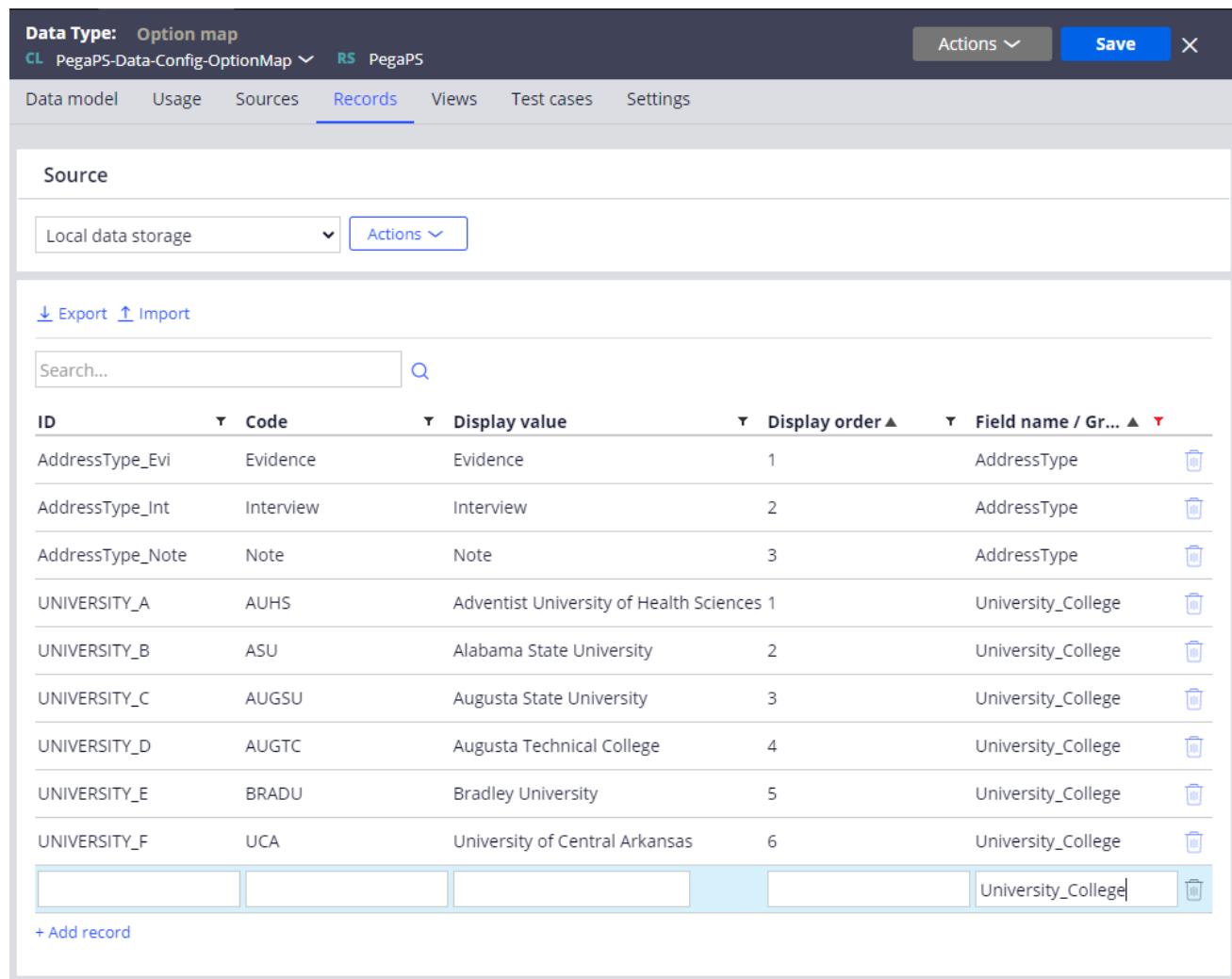
### Data type definition

Label	Property	Description
-------	----------	-------------

ID	.pyID	Unique Instance ID
Display Value	.pyLabel	Display Name
Code	.Code	Value
Display Order	.DisplayOrder	Integer index
Field Name / Group	.Purpose	Field Group (Category)

## Option Map Data Management

The OptionMap data type is created under Pega's local Data Type Wizard and can fully leverage the App Studio Data Designer for instance management and record editing.



The screenshot shows the 'Records' tab selected in the Data Type navigation bar. The main area displays a table of records with columns: ID, Code, Display value, Display order, Field name / Gr..., and a delete icon. The table lists various university and college names with their corresponding codes and display values. At the bottom, there is a row for a new record with empty input fields and a 'University\_College' dropdown. A note at the bottom indicates that the field name is derived from the code.

ID	Code	Display value	Display order	Field name / Gr...
AddressType_Evi	Evidence	Evidence	1	AddressType
AddressType_Int	Interview	Interview	2	AddressType
AddressType_Note	Note	Note	3	AddressType
UNIVERSITY_A	AUHS	Adventist University of Health Sciences 1		University_College
UNIVERSITY_B	ASU	Alabama State University	2	University_College
UNIVERSITY_C	AUGSU	Augusta State University	3	University_College
UNIVERSITY_D	AUGTC	Augusta Technical College	4	University_College
UNIVERSITY_E	BRADU	Bradley University	5	University_College
UNIVERSITY_F	UCA	University of Central Arkansas	6	University_College
				University_College

+ Add record

Option map data type record editor

## D\_OptionMapList Data Page

The D\_OptionMapList is a Requestor scoped data page that takes two parameters.

- FieldName (required) which supplies the label common to a collection of OptionMap instances.
- Code (optional) which may be provided in special circumstances to retrieve a specific OptionMap instance.

## LoadOptionMapList Data Transform

The D\_OptionMapList data page is populated by the LoadOptionMapList data transform which may be superseded or overridden in an implementation layer should detailed control of the list be required.

## Using OptionMap Data in a Picker control

The OptionMap data can be utilized in a Pega UI picker control by configuring the List source to call the D\_OptionMapList data page passing parameters to identify the field name/group that identifies the collection instances to be loaded.

The screenshot shows the configuration of a List source for a Picker control. The 'Type' dropdown is set to 'Data page'. The 'Data page' input field contains 'D\_OptionMapList'. A checkbox 'Search using data page parameter' is checked, with a note '(Recommended for large data sets)' below it. Below this, there is a table for parameters:

PARAMETER	VALUE
★ FieldName	"University_College"
Code	

*Option map auto complete list source*

# Product Overview

Pega Government Platform™ (PGP) provides a rich set of features designed to enhance the customer experience, improve user productivity, and increase customer satisfaction. The solution provides reusable components or processes that can be used to build an application or case by using Pega App Studio or Pega Dev Studio.

The following sections describe key functional areas and features of the application that you can use as-is or extend to meet your business needs.

- **Application intake features**
- **Manage information features**
- **Case Reviewer features**
- **Association Management Framework**
- **Entity Profile views**
- **Investigative case management**
- **Pega Government Platform case types and workflows**
- **Roles, portals, and dashboards in Pega Government Platform**
- **Primary data entities**
- **Time Management**
- **Further reading**

# Application intake features

Government agencies provide different services catering to their constituents. Constituents can be a person, business or any entity. PGP provides the following application intake-oriented features:

## Basic person information

A process that captures basic person information, validates it, and checks for the presence of duplicate information.

Multiple entity component can be used to capture information of multiple person entities as part of a single step in any case type.

## Detailed person information

A process that captures detailed person information like ethnicity, race, communication preferences, and so on, validates it, and checks for the presence of duplicate information.

## Business information

A process that captures business information like legal name, trade name, office location, communication details, and so on, validates it, and checks for the presence of duplicate information.

## Facility information

A process that captures facility information like facility name, contact details, location, and so on, validates it, and checks for the presence of duplicate information.

## Vehicle information

A process that captures vehicle information like model, make, registration number, garage location, owner details, and so on, validates it, and checks for the presence of duplicate information.

## Household information

A process that captures household information like household name, type, description, location, members of the household, and so on.

## Persist object component

A generic component that saves entity related information and data object information in the system of record.

## Capture experience

A process that captures the experience details of a person.

## Capture training

A process that captures the training details of a person.

## Capture location

A process that captures the location details of an entity.

## Consent agreement

Consent agreement is a feature that displays an actionable consent agreement screen. This screen contains customizable text for the agreement clauses of your organization, a check box to signify agreement, and a field to provide a signature. Pega Government Platform extends Pega App Studio to provide additional business configuration capabilities for the consent agreement.

## Item Information

A process that captures detailed Item information like Manufacturer, Owner, Dimensions, subitems, and so on, validates it, and checks for the presence of duplicate information.

## Basic Item Information

A process that captures basic person information, validates it, and checks for the presence of duplicate information.

# Manage information features

Use the following features to manage information about entities related to any government case.

## Manage education

A process that manages the education details of a person and stores those details in the system of record.

## Manage experience

A process that manages the experience details of a person and stores the details in the system of record.

## Manage training

A process that manages the training details of a person and stores the details in the system of record.

## Manage location

A process that manages the location details of an entity and stores the details in the system of record.

## Manage facility

A process that manages the facility entity details and stores the details in the system of record.

## Manage vehicle

A process that manages the vehicle entity details and stores the details in the system of record.

## Manage household

A process that manages the household entity details and stores the details in the system of record.

## Manage household members

A process that manages the members of household entity. When you add a new member or remove an existing member from a household, a new household is created, and the existing household becomes inactive.

## Manage item

A process that manages the item details and stores the details in the system of record.

## Manage Basic Item

A process that manages the item basic details and stores the details in the system of record.

## Manage business

A process that manages the business details and stores the details in the system of record.

## Manage person with basic details

A process that manages the person basic details and stores the details in the system of record.

## Manage person with complete details

A process that manages the person details and stores the details in the system of record.

## Manage multiple persons

A process that manages the multiple person details and stores the details in the system of record.

## Time entry

A process that logs time entry of a user against any case.

## Program configurator

A Program configurator is a tool allowing agencies to manage and customize the needs of the customer for programs like regulatory and integration, Citizen services, immigration-oriented programs more effectively and efficiently by interacting with constituents and corporate entities. The program Configurator software gives you greater flexibility and reduces the proliferation of items for uniquely configured program.

- A program configurator allows you to make use of the PGP application features with the ability to Configure profiles for individual constituents and corporate entities. It also allows you to configure the PGP Case Reviewer Features to help the application request case type to review the program throughout the life cycle. These case – review features such as Upload and verify documents, review checklist, evaluate to name a few aids the case review process that can be reused and customized in the application request case type to meet your business needs.
- The application request case type enables users to apply for any program or any of its descendants and takes you through steps to achieve end to end process of intake vs the review of the program so configured earlier. The application request case type allows business users to streamline application processes such as capturing image of a program, entity details, enables you to apply for a program

or any of its descendants, for example, a license. Using application request, you can bring all the services of your organization under one case type. The application request case type allows business users to streamline application processes, such as capturing entity details, document intake, consent agreement, and so on from start to finish. The business managers can review and approve the attached or uploaded documents, review the checklist, evaluate the process, and approve the application.

- The program configurator has an ability to view the programs and its descendants with separate views. To save time, you can reuse dynamic views, which are standard forms that your application automatically manages for each program that you define.

As a part of the program configurator a user can create a License configuration a specialized version of a program configurator by Providing data to set up a new license. License Configuration is a Comprehensive and configurable application that streamlines the licensing and certification application process from start to finish, including evaluations, document upload and its verifications.

Program configurator provides users with the following capabilities:

- Provide seamless and simple processes for constituents, businesses, and agency employees
- Reduce processing and minimize touch points to provide straight-through processing
- Provide a modern government platform to meet future requirements and allow business users to make policy changes
- Reduce project implementation time and accelerate project delivery
- Improve communication and interaction with constituents and businesses

## Case Reviewer features

Government case workers have various needs when reviewing the case through its lifecycle. Pega Government Platform provides the following features to aid the case review process that can be reused and customized to meet your business needs.

## Upload document component

Upload document component allows users to upload documents as part of any case. This component enables business users to dynamically configure the documents to be uploaded for any case with configurations like mandatory or non-mandatory documents, size of the documents.

## Verify document component

Verify document component allows the users to view the list of the uploaded documents and provides an option to approve or reject the documents. It also enables the business users to configure the change of stage on rejection.

## Nearby location component

Nearby location component allows users to view the list of nearby cases, nearby places and the nearby entities that are issued on a map based on the radius input calculated based on the location address provided in the application. The displayed results can be selected for further processing. This component can be added as a step in any case type and the location source, nearby results can be configured easily from App Studio

## Evaluate process

Evaluate process helps an evaluator or reviewer evaluate the application against various criteria that is important to your organization. After the application is evaluated, the system calculates a total evaluation score based on a weighted formula and recommends one or more next actions to take.

Pega Government Platform extends Pega App Studio to provide additional business configuration capabilities for evaluation criteria.

## Review checklist

Review checklist is a process through which a reviewer can review the application against various checklist items. The progress can be tracked by selecting the check boxes that correspond to items as they are completed.

Pega Government Platform extends Pega App Studio to provide additional business configuration capabilities for review checklist.

## Add interview

Add interview is a component that allows users to add interviews by selecting question templates, entering questions and responses in text areas, and adding participants, locations, attachments, and other details.

## Add evidence

Add evidence is a component that allows users to add evidence details, such as category, description, attachments, captured by, and location.

## Assessment

The Assessment configuration feature enables the user to configure the SLA score against each response and interpretation based on the final score.

Conduct assessment enables the user to conduct assessment in any case as a local action and go through the process of completing the answers for the assessment questions, save and return to complete the assessment. Also, on completion view the score, interpretation of the assessment.

Assessment component can also be used as a step in any of the stages of a new case type.

## Activity plan

An Activity Plan is a framework for tracking Goals and Tasks assigned to operators associated to an instance of a Case. The Activity plan component can be applied to various business use cases, including procurements, and investigation applications.

### Activity plan configuration

Activity plan templates expedite the process of creating a new activity plan. The templates are pre-configured activity plans containing all the details required to implement the activity plan. Activity Plan templates may be used to construct reusable collections of goals and tasks. A reusable template may be associated with a Case, and its goals and tasks may be configured for the specific Case.

Activity plan configuration enables users to configure activity plan templates that consist of goals, tasks, and SLA definitions. The user's Activity plan edit accessibility helps the user to update the Goals and tasks and sequence them based on priorities using the SLA's. Also, the existing activity plans can be deactivated if the template is not required.

### Activity plan template

An activity plan template Landing Page enables users to create activity plan templates with goals and tasks as a template which can be used for further processes. The users can edit the goals and tasks which can be further used in the investigation case type either by selecting the existing or creating a custom activity plan for the same. The users can also deactivate an Activity plan if the template is no more required. An activity plan so launched in the investigation case type can be appended with additional goals and tasks. This allows the users to add existing Goals or tasks, or custom create them.

### Initiation and execution

The component provides support for initiating the activity plans using the existing templates or by creating a custom template based on the use case. The system allows you to make necessary changes to the existing plan before initiation. It also provides

the option to assign the goals and tasks to the individual users by selecting from the existing list of users.

For any of the existing cases, activity plan can be initiated through local and case-wide actions and the case owner can assign tasks, goals to the team members and the execution can be monitored as per the configuration. On completion of all the tasks and goals, within an activity plan, the plan gets resolved. Also, the case owner can resolve the activity plan by forced closure if it is not required or is invalid.

### Activity plan usage

The Activity Plan Usage tab displays the list of plan templates that are used during creation of Activity plan case from Investigation or any other case. This tab also provides details on usage count, status of the plan and if the plan used is existing or custom. Users also have an option to convert custom plan to template.

### Entity merge

Entity merge is a feature that enables the users to merge the data properties of potential duplicates of Person entity or Business entity and create a single Master record. The duplicate entities are marked as inactive and all the properties selected by the user are retained in the Master entity.

The properties that need to be displayed for merging can be pre-configured by the users from option provided under PGP configuration in App Studio.

### Entity attachment for entities

The attachment component feature saves file and URL attachments as data items to the entities. It supports attaching multiple file types, such as Word, PDF, Excel, and so on. The features serve as tools to help manage critical attachments stored against any entity, such as person, business, facility, vehicle, and so on.

## Mobile Offline features

Mobile Offline features such as conduct interview, evidence, case notes and completion of the tasks in activity plan can be completed on mobile in offline mode. The data gets synced with the case once the users are online again.

## Internationalization

Internationalization enables a business architect or a supervisor user to configure the list of countries, the respective state or province labels that an application supports through a simple configuration. It also enables the phone number format validation for different countries.

## Case report

A case report is a detailed report of the data that is captured in the case type. You can now generate a Report of Investigation in both PDF and Microsoft Word formats. You can also incorporate tabular data into a Report of Investigation.

# Association Management Framework

Pega Government Platform provides a generic association framework that allows you to associate objects to each other and manage all the associations that are linked to the entities in the system.

## Person to Business, Person to Facility, Person to Vehicle, Person to Item

Ability to manage all the associations that are linked to a person entity in the system.

## Business to Facility, Business to Vehicle, Business to Person, Business to Item

Ability to manage all the associations that are linked to a business entity in the system.

## **Investigation to Subject, Investigation to Event, Investigation to Investigation**

Ability to manage all the associations that are linked to an investigation case in the system.

## **Subject to Person, Subject to Event, Subject to Investigation**

Ability to manage all the associations that are linked to a subject case in the system.

## **Entity Preview for Association Widgets**

This feature enables a case worker to preview entities associated. This feature allows to quickly preview the content on a separate panel without opening it. This allows you to display the most important information, so users can navigate and manage their entities much more efficiently.

## **Entity Profile views**

Pega Government Platform provides enhanced profile views that contain comprehensive information about the entities like person entity, business entity, vehicle entity, household entity, facility entity, item entity, and their related cases as well as associations. The views are built of different widgets that allow the customer to remove, re-arrange or add more widgets in similar patterns.

### **Enhanced person profile view**

PPega Government Platform provides agencies with the ability to create, update, and view profiles for individual constituents in the form of editable widgets, which provide the ability to edit existing information. Each profile includes personal information, demographics, associations, employment, history, and communication details..

**Morgan Consolidated**

**Details**

Trade name	Morgan Consolidated
Legal name	Morgan Consolidated
Employer ID	E132
Ownership type	Corporation

**Locations**

Map showing locations such as Ginger's Plumbing and Joga Bonito Soccer Club.

**Branches (0)**  
No data

**Associations (1)**  
Theft at White house museum  
Related

**Attachments (0)**  
No items

## Enhanced Business profile view

Pega Government Platform provides the ability to create, view, and manage business entities and the details of their employees, branches, associations to constituent profiles, and communication details in the form of editable widgets.

**Morgan Consolidated (BUSINESS-104)**

**Morgan Consolidated**  
Morgan Consolidated E132  
(123) 456-7890 — Morgan.Consolidated@pegacom  
557 N Crescent Street Dothan, Illinois 78786 US

**My cases**

Case name	Case ID	Case status
Manage associations	MA-3015	Resolved-Completed
Manage associations	MA-11	Resolved-Completed

**Timeline** View all

**Primary contact**   
Business email: Morgan.Consolidated@pegacom  
Business phone: (123) 456-7890  
Contact role: —  
Primary contact: Mr Sai Shankar  
Contact email: Morgan.Consolidated@pegacom  
Emergency phone: —

**Locations**

- 557 N Crescent Street Dothan Illinois 78786  
United States  
Mailing address  
Mapped
- 557 N Crescent Street Dothan Illinois 78786  
United States  
Office address  
Mapped

**Associations**

- Morency Matthew (Employee)
- 2011 Hyundai i20 (Owned • Jul 1, 2019 onwards)

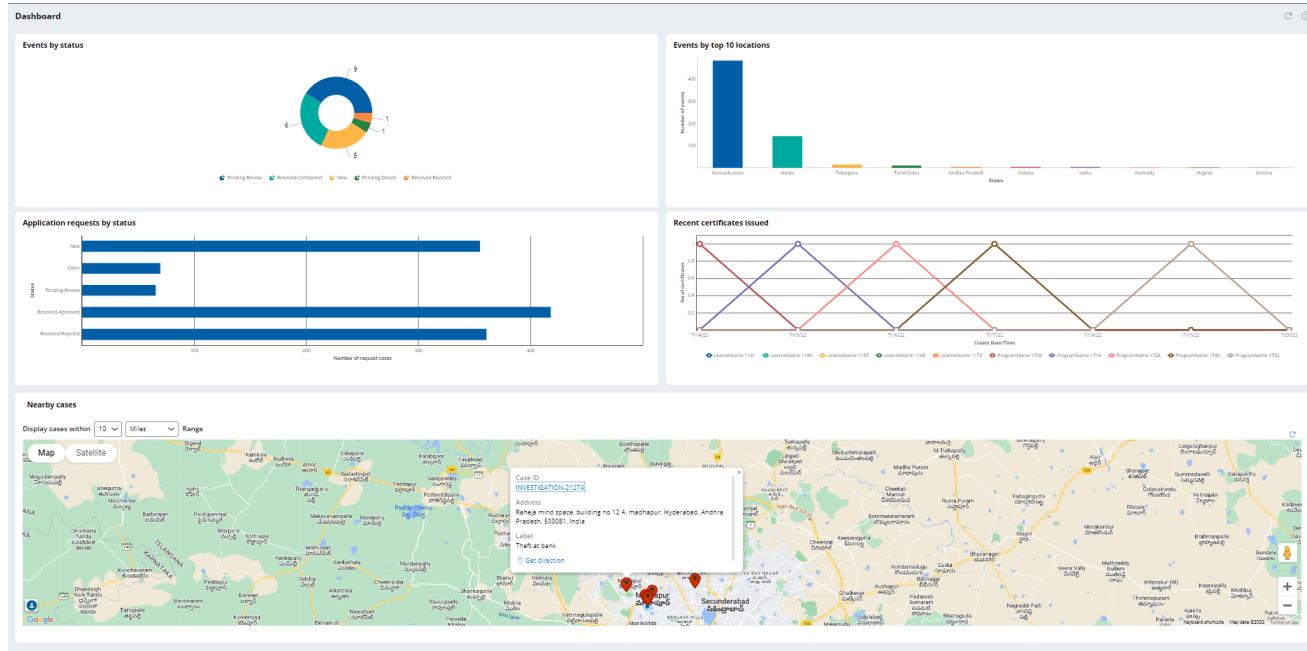
## Investigative case management

The Investigative Case Management (ICM) application has been built on Cosmos theme with the following enhancements:

- Event case has been converted to different stages and steps
- Investigation case has been converted to different stages and steps
- In the Event case type, feature of identifying the potential duplicate/related events has been introduced
- In Event case type, for Child welfare event type, new field to capture child details has been introduced and it creates a new subject from event review and associates to the investigation case

As part of this release, custom access in the request access feature is not supported in Cosmos and will be taken as part of next release.

## Enhanced view of ICM Application



## Event Case view

## Conduct Investigation

The Investigation case type provides end-to-end investigative case management abilities that enable agencies to increase the efficiency of the investigation's management and resolution process. The Investigation Case Management case type can be initiated as a standalone case type and also extends the Event case type. This event case type specializes in initiating an investigation case allows field investigators to accept ownership of the case and conduct the investigation.

A field investigator can then conduct the investigation with many tools such as subjects, Interview, evidence, Activity Plan, Assessment to name a few. By using a quick-create utility designed for speedy data capture, field investigators can establish and update cases. The drag-and-drop abilities of Pega Government Platforms enable the association of entities and situations for better analysis and visualization.

The Investigation case type also allows the field investigator to generate a report at the end of the investigation. This allows the investigator to capture the data captured using different tools used to conduct investigation.

At the completion investigation case by captures details using different tools, the field investigator can complete the stage to review the checklist and change the status of the case accordingly.

The investigation case type provides user to log Time for the case and provides investigation owner, supervisor with time tab which gives a summary of time entry logs of the case.

## Investigation Case view

## ICM Report of Investigation

The case report component enables you to generate a report of a case which contains the data that is captured using various other components that are available in the system. This helps you analyze the information from a single place.

You can generate a Report of Investigation in both PDF and Microsoft Word formats. You can display the data such as external organization, subject, evidence, and assessments from various sections in a tabular format. The ROI component is flexible,

allowing you to choose to include the data such as external organization, evidence, subject, interviews, assessments, and digital signature from various sections to be a part of the end report.

## Generate case report

## Generate document

To make changes, edit the details and continue to attach the document to the case.

**Case summary**

**Details**

Investigation: Theft at White house museum  
 Investigation ID: INVESTIGATION-110097  
 Category :Theft  
 Description: Theft at White house museum  
 Case open date: July 20, 2022, 02:45:00

**External organizations**

Employer ID	Legal name	Trade name	Primary contact	Email	Phone
E132	Morgan Consolidated	Morgan Consolidated	Sai Shankaran	Morgan.Consolidated@pegacom	(123) 123-1231
PBS1231	PBS1	PBS	Krishna Chaitanya	Chaitanyasdfasdfa@pbs.org.com	

**Evidence**

Evidence ID	Evidence type	Short description	Details	Captured by	Associated Subject	Captured Date	Location
EVIDENCE-93110	Anecdotal	CCTV footage		ankig	July 20, 2022, 02:49:00	Madhapur Telangana 500081 India	
EVIDENCE-93111	Character	Knife		ankig	July 20, 2022, 02:50:00	Madhapur Telangana 500081 India	
EVIDENCE-93112	Circumstantial	Medical Kit		ankig	July 20, 2022, 02:50:00	Madhapur Telangana 500081 India	

**Associated subjects**

Subject ID	Name	Description	Role	Entity Type	Entity ID	Updated date
SUBJECT-107111	Marty Solomon	Marty Solomon	Person of interest	Person	PERSON-102027	July 20, 2022 02:49:49

**Team**

Name	Role	Email
Ankireddy, Gurusai Ravi Raja Reddy	Investigation owner	GurusaiRaviRajaReddy.Ankireddy@in.pega.com

**Activity plans**

**Activity plan: Investigation plan**

Description: Investigation plan description

**Plan tasks**

Name	ID	Description	Assigned to	Due date
Collect evidence	TASK-182902	Collect evidence description	ankig	



**Generate case report**

- 11. Did you tell anyone about the incident or behavior? Who?
- 12. Do you know why the incident or behavior occurred?
- 13. Do you know anyone else who can shed light on this incident?
- 14. Is there anything else you want to tell me that I haven't asked you?

**INTERVIEW-92096**

Interview credibility: High  
 Interview template: Subject  
 Interviewed By: Ankireddy, Gurusai Ravi Raja Re+  
 Interview start time: July 20, 2022, 02:52:00  
 Interview end time: null  
 Location: Madhapur, Telangana, 500081, India

**Interview notes**

- 1. What happened?
- 2. Is there any reason anyone would invent or lie about the incident?
- 3. Where were you when the alleged incident occurred?
- 4. Do you have any witnesses who can corroborate your whereabouts at the time of the incident?
- 5. When and where did this happen?
- 6. What were the circumstances leading up to the incident?
- 7. Who else was involved?
- 8. What is your connection to the complainant?
- 9. Are you aware of any other complaints by this person?
- 10. Recount the dialogue that occurred in order of what was said.
- 11. What did the complainant do or say?
- 12. Is there any evidence to support your account of what happened?
- 13. Is there anyone else we should talk to who had knowledge of the incident or the circumstances surrounding it?
- 14. Have you talked to anyone about the incident? Who? What did you tell them?

Capture digital signature?

I approve the contents of this report as shown.


[Back](#)[Continue](#)*Investigation report*

## Operator Case Involvement (Touch) Reporting

This feature enables the operators to create reports and expose them to supervisors. Each operator can view the touch point reports and view their course of work over the period of time. These reports are only restricted to the work done in the Investigation case type.

These reports allow you to monitor in real-time the efficiency and success. You can view how often an operator has worked on an investigation case type and how long it takes to complete the tasks on an average, for example, over the last year.

A supervisor can generate the following reports:

- A touch point report for the supervisors to view their work.
- A supervisor report for touch points of team for investigation to view the work done by the team members on the investigation case type.

## Pega Government Platform case types and workflows

This section describes the standard cases that are included with Pega Government Platform and are available for selection within the New Application wizard when you create your Pega Government Platform application. Depending on your organization's needs, you may include all or a subset of these case types in your application.

- [Example case types](#)
- [Standard case types](#)

### Example case types

#### Event case management

Pega Government Platform includes end-to-end event case management that enables organizations to improve the resolution process. It provides the following capabilities:

- Initiate events
- Review events
- Resolve events

#### Investigative case management

Pega Government Platform includes an end-to-end investigative case management application that enables organizations to improve the efficiency of the investigations management and resolution process by reducing paperwork and providing a digital platform. It provides the following capabilities:

- Event intake – enables the intake worker to capture event details and category, and to send it for review.
  - Initiate investigation cases – enables the ICM agents and supervisors to initiate investigation cases through an event or standalone cases.
  - Case acceptance through email – enables the ICM agents to accept the investigation cases directly through E-mail.
  - Subjects case management – enables the ICM agents to create subjects, and to associate subjects to investigation cases or known entities.
  - Conduct interviews – enables the ICM agents to conduct interviews on subjects and to add interviews to an investigation case.
  - Capture evidence – enables the ICM agents to add evidence and capture all the details for an investigation case and associate it with subjects.
  - Add case notes – enables the ICM agents to add case notes that are related to any investigation to the investigation case.
  - Add activity plan – enables the ICM case owner to add an activity plan to an investigation case and assign tasks to the team members
- 
- Create assessment – enables the ICM agents to conduct assessments for entities that are related to any investigation case.
  - Quick creates – enables the ICM agents to create subjects and investigation cases on mobile and hand-held devices.
  - Generate report of investigation – enables the agents and supervisors to generate a report of an investigation by selecting the contents to be included in the document.
  - Team management – enables the ICM agents (case owners) to manage the team members for an investigation case.
  - Manage associations – enables the agents and supervisors to manage all the associations that are related to an investigation case, subject case, event case, and entities.
  - Persona based dashboards – ICM agents, supervisors, and intake specialists have specific dashboards that enable them to track events, investigations, team, work, and workload.

- Access management – enables the ICM agents to request read and write access, and to approve access as read, write, or custom access.
- Case and object visualizations – enables the view of all the associations in a visual representation, which gives a 360-degree view of all the tied objects.
- Audit trails and timeline views – enables the view of the activities in any case in a visual timeline view or list view.
- View of investigation case – enables the view of all details related to an investigation case in a 360-degree UI to get a complete picture of the case.
- Transfer ownership – Investigation owner can transfer ownership of an investigator to an office or an user based on the requirement. A user has the option to view the investigation details and can accept the ownership or reject it so that the investigation can retransfer the ownership to someone else.

## Complaints management

Pega Government Platform includes an end-to-end complaints management application which is available only in UI Kit version. It provides the following capabilities:

- File complaints – enables constituents to create complaint cases and provide all related details.
- File anonymous complaints – enables constituents to file an anonymous complaint.
- Review complaints – enables managers to review the complaint cases and take necessary action.
- Persona based dashboards – Complaints workers and managers have specific dashboards that enable them to view complaint status, workload, team members, and work lists.

## Procurements management

Pega Government Platform provides a case management-based application for managing the procurement process which is available only in UI kit version. It provides the following capabilities:

- Define specifications – enables the procurement officer to define specifications for the new procurement.
- Add marketing details – enables the market analyst to add and capture details for the marketing related to new procurement.
- Define advertisements – enables the procurement officer to define advertisement details for a procurement.
- Vendor bidding – enables the vendors to submit bids against the listed procurements.
- Multi-personnel evaluation – enables multiple levels of evaluation for the submitted bids.
- Award bid process – enables the supervisor to award a bid based on the evaluation score and system recommendation.
- Ask questions – enables the end users to submit questions.
- View answers – enables the end users to view the existing questions and answers on the portal.
- Persona based dashboards – Procurement officers, supervisors, and market analysts each have specific dashboards that enable them to view procurement status, team workload, team members, and work lists.

## User registration

Constituents can register and obtain an online account through Pega Government Platform to access services provided by agencies without the need to come into an office or call the agency. After registering, users can manage their own operator ID and password. The user registration process utilizes Pega's authentication and authorization services, which you can integrate into your agency's security infrastructure.

## Standard case types

Case Type	Description
Add person	Add details of a person entity and create a person entity in the system

<b>Case Type</b>	<b>Description</b>
Add business	Add details of a business entity and create a business entity in the system
Add facility	Add details of a facility entity and create a facility entity in the system
Add vehicle	Add details of a vehicle entity and create a vehicle entity in the system
Add household	Add details of a household entity and create a household entity in the system
Add evidence	Add details of evidence in the system
Add note	Add details of note in the system
Add activity plan	Create an activity plan case in the system
Activity plan template	Allows users to manage the activity plan template configurations
Add item	Add details of an item entity and create an item entity in the system
Application Request	A process to file an application. For entities, to request issue of a program or license
Conduct interview	Add details of interview case in the system
Goal	Create goal case in the system
Initiate event	Add information about an event and create a new event case in the system
Manage associations	Allows users to manage all the associations that are linked to the entities in the system
Initiate complaint	Add information about a complaint and create a new complaint in the system
Initiate procurement	Add information about a procurement and create a new procurement in the system

<b>Case Type</b>	<b>Description</b>
Initiate investigation	Add information about an investigation and create a new investigation in the system
Initiate subject	Add information about a subject and create a new subject in the system
Change password	Provides an option to change the existing password
Forgot password	Provide an option to retrieve a forgotten password
License	Add details of license and persist to the system
Program	Add details of program and persist it to the system
Question	Provide an option to the users to ask questions
Registration	Create and register new user account in the system
Transfer ownership	Provides a capability for transferring the case ownership to other users
Request access	Enables the users to request access to a case which is routed for approval
Merge Entity	Provide an option to merge the potential duplicates of Person Entity and/or Business Entity
Manage Branches	Manage the branches of business entity and save it to the system
Manage Education	Manage the education details linked to person entity and save it to the system

<b>Case Type</b>	<b>Description</b>
Manage Experience	Manage the experience details linked to person entity and save it to the system
Manage Location	Manage the location linked to any entity and save it to the system
Manage Training	Manage the training details linked to person entity and save it to the system
Manage household members	Manage the household members linked to household entity and save it to the system
Subject	Add details of a subject entity and persist it to the system
Search Person	Search a person entity in the system based on search criteria like id, name
Search Business	Search a business entity in the system based on search criteria like id, name
Search Household	Search a household entity in the system based on search criteria
Search Event	Search an event in the system based on search criteria like id, name
Search Facility	Search a facility entity in the system based on search criteria like id, name
Search Vehicle	Search a vehicle entity in the system based on search criteria like id, name
Search Complaint	Search a complaint in the system based on search criteria like id, name
Search Investigation	Search an investigation in the system based on search criteria like id, name

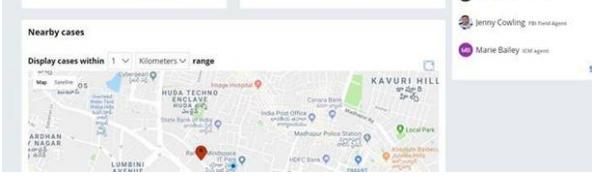
<b>Case Type</b>	<b>Description</b>
Search Procurement	Search a procurement in the system based on search criteria like id, name
Task	Create a task case in the system
Time entry	Add details of a Time entry such as time code, project code, reference id, hours, and so on and persist it to the system
Update person	Update the details of person entity and store in the system
Update business	Update the details of business entity and persist it to the system
Update facility	Update the details of facility entity and store in the system
Update item	Update the details of item entity and persist it to the system
Update vehicle	Update the details of vehicle entity and store in the system
Update household details	Update the details of household entity and store in the system
View person	View all the information of person profile in a 360-degree view
View business	View all the information of business profile in a 360-degree view
View household	View all the information of household entity in the system
View item	View all the information of item profile in a 360-degree view .

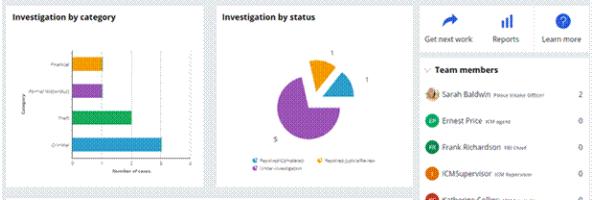
Case Type	Description
View vehicle	View all the information of a vehicle entity in the system
View facility	View all the information of a facility entity in the system
View complaint	View all the information of a complaint in the system
View investigation	View all the information of an investigation in the system
View procurement	View all the information of a procurement in the system

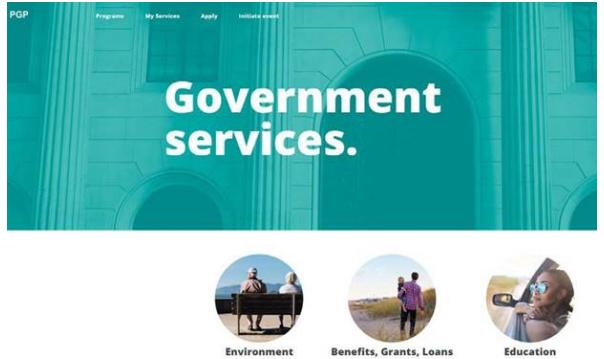
## Roles, portals, and dashboards in Pega Government Platform

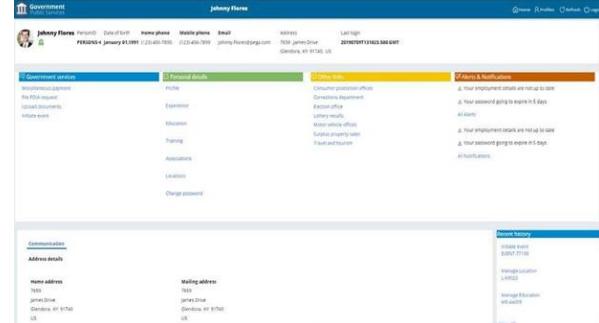
Pega Government Platform supports key roles associated with your day-to-day use of the application. The table below describes the key access roles provided with the Pega Government Platform.

Role name	Description
PGP User	Pega Government Platform users can view urgent work, add new entities to the system, manage the entity information, and view reports from the PGP Case worker dashboard via the Pega manager portal.
PGP Manager	The manager views urgent work, and reviews and approves workbasket items from the PGP manager dashboard which serves as a generic portal for any apps built on top of

Role name	Description
	PGP. The manager also has an ability to configure assessments.
Intake specialist	The intake specialist creates events based on reported information and views reports about events intake. They use the templates of the PGP Case worker dashboard and Pega manager portal with reports specific to intake.
ICM agent	<p>The ICM agent works on the investigation cases as the investigator or investigator analyst and uses the templates of the PGP Case worker dashboard and Pega manager portal with reports specific to investigations.</p> 
ICM supervisor	<p>The ICM supervisor reviews events and assigns work to the ICM agents and uses the templates of the PGP Manager dashboard and Pega manager portal with reports specific to investigation case summaries. Supervisor also has an ability to configure assessments.</p>

Role name	Description
Procurement Officer	 <p>The procurement officer creates the procurement plan, enters advertising details, and answers vendor questions about the procurement. They use the templates of the PGP case worker dashboard and Pega manager portal with reports specific to procurements cases.</p>
Procurement Specialist	<p>The procurement specialist enters specifications and evaluation criteria for the procurement and reviews the procurement. They use the templates of the PGP Manager dashboard and Pega manager portal with reports specific to procurements cases.</p>
Procurement Supervisor	<p>The procurement supervisor reviews the procurement plan details and approves/rejects them. They use the templates of the PGP Manager dashboard and Pega manager portal with reports specific to procurements cases.</p>
Market Analyst	<p>The market analyst enters market analysis details about the procurement using the templates of the PGP Case worker dashboard and Pega manager portal.</p>

Role name	Description
Procurement Manager	<p>The procurement manager awards the procurement and oversees the work of the entire team using the templates of the PGP Manager dashboard and Pega manager portal.</p>
Vendor	<p>The vendor submits bids against a procurement. Vendors use the self-service portal.</p>
Complaints User	<p>The complaints user can file a complaint from their self-service portal as a constituent.</p>
	
Complaints Manager	<p>The complaints manager reviews filed complaints and takes further action as well as coordinates work for his or her team. They will use the templates of the PGP Manager dashboard and Pega manager portal.</p>
PGP End User	<p>The PGP end user can view and manage personal details like experience, education,</p>

Role name	Description
	<p>associations, training, address. They can also initiate events from the self-service portal.</p> 

## Primary data entities

This section describes the data types that are used by Pega Government Platform, stored in external systems or, where appropriate, internally to the Pega system.

Name	Description	System of record (SOR)
Address	Contains information pertaining to the address details of a person, business, facility, vehicle, household, and associate.	Pega
Association	Contains information pertaining to managing all associations.	Pega
Assessment	Contains information pertaining to the configurations and creation of assessment	Pega

Name	Description	System of record (SOR)
Checklist Map	Contains information pertaining to the checklist items to be reviewed for various applications.	Pega
Communication	Contains information pertaining to communication details, such as email, fax, phone, social network, and websites.	Pega
Complaint	Contains attributes and information pertaining to complaints filed by constituents or businesses.	Pega
Configuration	Contains information pertaining to the configurations of an address, country, state, authorization, and communication type.	Pega
Document	Contains information pertaining to the document intake and its verification	Pega
Education	Contains information pertaining to the education details of a person.	Pega

Name	Description	System of record (SOR)
Entity	Contains data pertaining to a business, associate, person, vehicle, facility, household, item, and place.	Pega
Experience	Contains information pertaining to the experience details of a person.	Pega
Evaluation Criteria	Contains information pertaining to the evaluation criteria and evaluation scores for different applications.	Pega
Event	Contains information pertaining to an event or an incident.	Pega
Evidence	Contains information pertaining to evidence.	Pega
Goal	Contains information pertaining to goal.	Pega
Identifier	Contains information pertaining to identifiers such as passport, driver license, SSN, or national ID.	Pega
Interview	Contains information pertaining to an interview.	Pega

Name	Description	System of record (SOR)
Investigation	Contains information pertaining to an investigation.	Pega
License config	Contains information pertaining to license configuration	Pega
Link	Contains information pertaining to rules related to linked data objects.	Pega
Note	Contains information pertaining to case notes.	Pega
Plan	Contains information pertaining to activity plan	Pega
Portal	Contains information pertaining to the details to be displayed on the portal based on the entity type (Person or Business) and business requirement.	Pega
Procurement	Contains attributes and information pertaining to procurements.	Pega
Program Config	Contains information pertaining to program configuration	Pega
Project code	Contains information pertaining to project code	Pega

Name	Description	System of record (SOR)
Question	Contains information pertaining to questions submitted by end users.	Pega
Subject	Contains information pertaining to a subject.	Pega
Task	Contains information pertaining to task	Pega
Training	Contains information pertaining to the training details of a person.	Pega
Time code	Contains information pertaining to time code.	Pega
Time entry log	Contains information pertaining to time entry log submitted by users.	Pega

## Time Management

- [Time entry](#)
- [Manage Time](#)
- [Roles in Time Management](#)

### Time entry

This case type provides users with a process to log time with details like project code, time code, reference id (case id), hours, and so on. These logs are reviewed by the team manager in the review step.

If a user is logged into time for the first time (on the current day), they can copy the projects, activities, and references by selecting the latest entry check box. This action will copy the latest submitted time logs to the present-day logs.

A Touch point report that lists the cases along with their description is provided to the users when an action is performed.

## Manage Time

In My Timesheet users can search for time entry logs with search criteria such as start day, end day, project, activity, and reference. The reference id is provided as a link with which users can view the case for which they have logged time.

The screenshot shows the 'Manage Time - My Timesheet' interface. At the top, there is a search bar with fields for 'Start Day' (7/13/2022) and 'End Day' (7/20/2022). Below the search bar, there are four summary statistics: 'Total tracked 7.00 hours', 'Total approved 0.00 hours', 'Total pending approval 7.00 hours', and 'Total rejected 0.00 hours'. To the right of these stats are buttons for 'Group', 'Fields', 'Density', and 'Refresh'. Below the summary is a table with columns: Date, Project, Activity, Reference, Status, and Hours. The table contains the following data:

Date	Project	Activity	Reference	Status	Hours
Jul 15, 2022	Case Worker	Surveillance Observation		Submitted	1.00
Jul 20, 2022	Investigation Project	Report		Submitted	1.00
Jul 20, 2022	Investigation Project	Report	TE-37016	Submitted	2.00
Jul 20, 2022	Case Worker	Research	TE-37006	Submitted	3.00

*Manage Time - My Timesheet*

Team managers are provided with Team Timesheet along with My Timesheet. In Team Time sheet, managers can view the logs of team members all together or view the logs of specific team member with Operator search criteria.

**Manage time**

**TEAM TIMESHEET**

Operator: All

Start Day: 7/19/2022 End Day: 7/20/2022 Project: All Activity: All Reference: All

Total tracked: 20.08 hours Total approved: 9.00 hours Total pending approval: 11.08 hours Total rejected: 0.00 hours

Clear all filters Group Fields Density Refresh

Date	Team	Operator	Project	Activity	Reference	Status	Hours
Jul 19, 2022	Default team for PegaPS	Pudu, Alekhy	Case Worker	Research	TE-30007	Submitted	1.00
Jul 13, 2022	My team	Anirreddy, Gurusai Ravi Raja Reddy	Case Worker	Surveillance Observation	INVESTIGATION-106055	Approved	4.00
Jul 13, 2022	My team	Anirreddy, Gurusai Ravi Raja Reddy	Case Worker	Surveillance Observation	TE-35001	Approved	4.00
Jul 13, 2022	Default WorkGroup	Nemani, Nishitha	Case Worker	Surveillance Observation	INVESTIGATION-106056	Approved	1.00
Jul 15, 2022	CaseWorker	CaseWorker_Cosmos	Case Worker	Surveillance Observation	TE-36009	Submitted	1.00
Jul 15, 2022	Default WorkGroup	Srikarleni, Sreven	Case Worker	Surveillance Observation	TE-36009	Submitted	0.03
Jul 19, 2022	Default team for PegaPS	Pudu, Alekhy	Case Worker	Research	TE-30016	Submitted	3.00
Jul 20, 2022	CaseWorker	CaseWorker_Cosmos	Investigation Project	Report	TE-37016	Submitted	1.00
Jul 20, 2022	CaseWorker	CaseWorker_Cosmos	Investigation Project	Report	TE-37016	Submitted	2.00

1 2 Next

Manage time entry - Team timesheet

## Roles in Time Management

Time Management supports key roles associated with your day-to-day use of the application. The following table lists the key access roles with their descriptions.

Role name	Description
User	Users can log time entries and view their own logs. Managers can view their team's entry logs (if they managing any teams) along with their own logs.
Admin	Admin has access to view the time entries of all the users .

## Further reading

For more information about Pega Government Platform, see <https://community.pega.com/knowledgebase/products/government-platform>.

# Additional

Filler text.

- **Pega Certification and Licensing for Government hotfixes**

## Pega Certification and Licensing for Government hotfixes

The following tables list required hotfixes the Pega Certification and Licensing for Government application requires. To request a hotfix, go to [My Support Portal](#). Click New request > For something I need and select Service request > Existing hot fix. Add and verify the hotfix details and click Finish.

Import each type of hotfix in the listed order during the Pega Government Platform or Pega Foundation for Government installation or upgrade:

- Apply **Pega Platform** hotfixes immediately after the Pega Platform installation or upgrade.
- Apply **Pega Certification and Licensing for Government** hotfixes just after you complete the application bundle import.

To see hotfix installation details, see the readme that is included in the hotfix.

- [Hotfixes for Pega Certification and Licensing for Government 8.x](#)
- [Hotfixes for Pega Certification and Licensing for Government 7.x](#)

## Hotfixes for Pega Certification and Licensing for Government 8.x

## Pega Certification and Licensing for Government 8.5

The following table lists the required **Pega Certification and Licensing for Government 8.5** hotfixes.

Pega Platform Version	Hotfix Number	Observed Behavior
8.5.3	HFH-81747	Addressed the issues with FetchActiveLicenseCount report. Hotfix is provided to update the column case to match the case defined on external mapping for the classes.

## Pega Certification and Licensing for Government 8.4

The following table lists the required **Pega Certification and Licensing for Government 8.4** hotfixes.

Pega Platform Version	Hotfix Number	Observed Behavior
8.4	HFH-67424	In Pega Certification and Licensing for Government 8.4, security has been enhanced on a number of rules by the addition of an existing privilege "AllFlows".

## Pega Certification and Licensing for Government 8.3

The following table lists the required **Pega Certification and Licensing for Government 8.3** hotfixes.

Pega Platform Version	Hotfix Number	Observed Behavior
8.3	HFx-67641	In Pega Certification and Licensing for Government 8.3, security has been enhanced on a number of rules by the addition of an existing privilege "AllFlows".

## Hotfixes for Pega Certification and Licensing for Government 7.x

### Pega Certification and Licensing for Government 7.31

The following table lists required **Pega Certification and Licensing for Government 7.31** hotfixes.

Pega Platform Version	Hotfix Number	Observed Behavior
8.1	HFx-49065	Not able to cancel the adjudication work object when it is created via summon case.
	HFx-47984	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• Status fail screen is shown on click of Create Operator link on the login screen.</li> <li>• Username and password text overlaps with the text input</li> </ul>

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>fields on the login screen.</p> <ul style="list-style-type: none"> <li>Click on Finish button in Apply for License process does not trigger an action.</li> </ul>
7.4	HFx-49065	<p>Not able to cancel the adjudication work object when it is created via summon case.</p>
	HFx-43465	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>Processing error while creating 'Apply for exam' flow.</li> <li>Create Operator link is not visible properly.</li> </ul>
7.3.1	HFx-49065	<p>Not able to cancel the adjudication work object when it is created via summon case.</p>
	HFx-47490	<p>Next Category button is displayed even when no categories exist.</p>
	HFx-46670	<p>The below properties are missing for the Data page D_CitizenProfile: Internal</p>

Pega Platform Version	Hotfix Number	Observed Behavior
		Notes, Personal Address Type, Person Citizenship Text, State Residence Indicator.
	HFix-46482	Error occurs upon running the D_StateList datapage.
	HFix-46034	Property validation errors are observed when testing the database connection for the PegaGov-Data-CitizenProfile class.
	HFix-43465	Changes are required to address the following issues: <ul style="list-style-type: none"> <li>Processing error while creating 'Apply for exam' flow.</li> <li>Create Operator link is not visible properly.</li> </ul>

## Pega Certification and Licensing for Government 7.21.01

The following table lists required **Pega Certification and Licensing for Government 7.21.01** hotfixes.

Pega Platform Version	Hotfix Number	Observed Behavior
7.3.1	HFix-40867	Address section misaligned when using OOTB entity address section.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-32809	Support for Pega Certification and Licensing for Government 7.21 release.
	HFx-35628	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>• The Create Operator page user interface is not working as expected.</li> <li>• Logging out from the PFG Operator page encounters an error page.</li> <li>• The Post-Suspension page contains a correspondence issue.</li> <li>• The Edit-Suspension process does not show correspondence.</li> <li>• Opening a complaint case from search complaint encounters an error.</li> <li>• The return to login page screen has a blank display.</li> </ul>

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> <li>Users cannot view the Add business button in the Create operator page.</li> </ul>
7.3	HFix-40867	Address section misaligned when using OOTB entity address section.
	HFix-32809	Support for Pega Certification and Licensing for Government 7.21 release.
	HFix-35628	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> <li>The Create Operator page user interface is not working as expected.</li> <li>Logging out from the PFG Operator page encounters an error page.</li> <li>The Post-Suspension page contains a correspondence issue.</li> <li>The Edit-Suspension process does not show correspondence.</li> </ul>

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> <li>Opening a complaint case from search complaint encounters an error.</li> <li>The return to login page screen has a blank display.</li> <li>Users cannot view the Add business button in the Create operator page.</li> </ul>
7.2.2	HFix-40867	Address section misaligned when using OOTB entity address section.

## Pega Certification and Licensing for Government 7.15.01

The following table lists the required **Pega Platform** hotfixes for **Pega Certification and Licensing for Government** 7.15.01.

Pega Platform Version	Hotfix Number	Observed Behavior
7.1.9	DL-50540, HFix-24020	Not able to Lock and save the ruleset.
	DL-50541, HFix-24403	Accessibilty Issue: Confirm Message is not the first message.

The following table lists the required Pega Certification and Licensing for Government 7.15.01 hotfixes.

Pega Platform Version	Hotfix Number	Observed Behavior
7.2.2	HFHix-26659	Need to lock the existing two open ruleset versions.
	HFHix-27191	Upgrades to Pega Certification and Licensing for Government 7.15 introduce incorrect database table name.
	HFHix-27532	Expertise section is not reachable for a constituent.
	HFHix-29612	Password mismatch error while creating operator.
	HFHix-29858	Missing class key for Address and CommOptions classes.
7.2.1	HFHix-25503	A voided License is not displayed as withdrawn for Pega Certification and Licensing for Government.
	HFHix-25625	No products are displayed for New-BEIssueLicense flow, when "Marine Harvesting" category is selected.
	HFHix-25889	Incorrect suspension type displayed for Summons Admin Process.
	HFHix-26074	Error came up after payment is submitted in Renew License flow.

Pega Platform Version	Hotfix Number	Observed Behavior
7.2	HFH-26659	Need to lock the existing two open ruleset versions.
	HFH-27191	Upgrades to Pega Certification and Licensing for Government 7.15 introduce incorrect database table name.
	HFH-27532	Expertise section is not reachable for a constituent.
	HFH-29858	Missing class key for Address and CommOptions classes.
	HFH-25503	A voided License is not displayed as withdrawn for Pega Certification and Licensing for Government.
	HFH-25625	No products are displayed for New-BEIssueLicense flow, when "Marine Harvesting" category is selected.
	HFH-25889	Incorrect suspension type displayed for Summons Admin Process.
	HFH-26074	Error came up after payment is submitted in Renew License flow.
	HFH-26659	Need to lock the existing two open ruleset versions.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-27191	Upgrades to Pega Certification and Licensing for Government 7.15 introduce incorrect database table name.
	HFx-27532	Expertise section is not reachable for a constituent.
7.1.9	HFx-26659	Need to lock the existing two open ruleset versions.
	HFx-27191	Upgrades to Pega Certification and Licensing for Government 7.15 introduce incorrect database table name.
	HFx-27532	Expertise section is not reachable for a constituent.

## Pega Certification and Licensing for Government 7.13.01

The following table lists required **Pega Certification and Licensing for Government 7.13.01** hotfixes.

Pega Platform Version	Hotfix Number	Observed Behavior
7.1.5	HFx-27808	Incompatible datatypes exist with Integration layer.
	HFx-29811	Additional rules required to support missing default value of FishingTime property in CLF framework.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFix-30129	Two map rules required for FishingTime property in CLF framework.