



Pega Foundation for Financial Services '23

4 March 2024

CONTENTS

Get started	12
Release notes	13
Pega Foundation for Financial Services '23 enhancements.....	13
Issues addressed in this version.....	15
Known issues in this version.....	16
Updating from previous releases.....	16
Deprecated and withdrawn rules.....	17
Deprecated rules.....	18
Withdrawn rules.....	61
Install	88
Install.....	88
Backing up your system.....	88
Completing the prerequisite tasks.....	89
Installing the application.....	90
Update and Hotfixes	95
Pega Foundation for Financial Services update guide.....	95
Backing up your system.....	95
Completing the prerequisite tasks.....	96
Updating the application.....	97
Pega Foundation for Financial Services hotfixes.....	103
Hotfixes for Pega Foundation for Financial Services 8.x.....	103
Pega Foundation for Financial Services 8.8.....	104
Pega Foundation for Financial Services 8.7.....	107
Pega Foundation for Financial Services 8.6.....	107
Pega Foundation for Financial Services 8.5.....	110
Pega Foundation for Financial Services 8.4.....	112
Pega Foundation for Financial Services 8.3.....	114
Pega Foundation for Financial Services 8.2.....	117
Pega Foundation for Financial Services 8.1.....	118

Hotfixes for Pega Foundation for Financial Services 7.x.	121
Pega Foundation for Financial Services 7.4.	122
Pega Foundation for Financial Services 7.32.	129
Pega Foundation for Financial Services 7.31.	132
Pega Foundation for Financial Services 7.22.	136
Pega Foundation for Financial Services 7.21.1.	140
Pega Foundation for Financial Services 7.21.	143
Pega Foundation for Financial Services 7.17.	144
Pega Foundation for Financial Services 7.16.	147
Pega Foundation for Financial Services 7.15.	149
Pega Foundation for Financial Services 7.14.	149
Product overview	151
Pega Foundation for Financial Services features.	151
Search and composite view.	152
Determining entity risk.	152
Information providers.	152
Onboard contracting party requirements.	153
Financial accounting.	154
Operational and product structures.	154
Financial services data.	155
Customer master profile.	156
Event driven architecture.	156
Async processing.	156
Roles, portals, and dashboards.	157
Primary data entities.	158
Further reading.	161
Implement	162
Pega Foundation for Financial Services implementation guide.	162
Preparing for implementation.	162
Prerequisites.	165
Creating new application.	165
Creating the baseline application.	165

Loading sample data.....	168
Defining the data model.....	168
Pre-configured data types.....	169
Mapping the application data.....	171
Access groups.....	172
Defining access roles and privileges.....	174
Defining your access groups.....	174
Configuring access groups.....	174
Adding an access group.....	175
Adding an access role.....	175
Connecting system of record.....	176
Integration layer (PegaFS-Int-).	176
Example: Investment security integration.....	177
Connect FS products to system of record.....	183
Before connecting to another system of record.....	184
Configuring the operating structure.....	184
Configuring the product structure.....	185
Data storage and structure of products.....	186
Sample products in FSF_PRODUCTMATRIX table.....	187
Class structure and rules.....	188
Connecting to another data source.....	190
References.....	191
Managing reference data.....	192
Application configurations.....	195
Updating the localization of currency fields.....	195
Integrations.....	198
Navigating to the pre-configured external data pages.....	199
Defining the security model and organization structure.....	200
Security model.....	200
Organizational structure.....	201
Defining your authentication scheme.....	205
Authentication schemes.....	205

Implementing your authentication scheme.....	206
Authorization scheme.....	206
Customizing the user experience.....	207
Features implementation.....	207
Financial Services core features.....	208
T.O.M - Reusability of org charts created by other methods.....	209
Requirements management.....	210
Accounting.....	210
Third party data providers.....	211
Data management.....	212
Externalization of third party data.....	214
Solution overview.....	214
Implementation.....	215
Overview.....	216
Externalization on save.....	217
Externalization on access.....	217
On-demand loading.....	219
Synchronization tuning.....	221
Cleanup of data from WorkFolder.pyWorkParty(Customer).....	224
Externalization of third party provider.....	224
Adding a new data source for entity enrichment.....	231
Adding a new component.....	232
Implementing the component.....	233
End results and Use case examples.....	238
References.....	240
Currency conversion structure.....	240
Soft delete functionality.....	242
Class structure.....	245
Soft-delete logic.....	246
Implementation in Financial Services (FS) applications.....	247
Operating structure management (T.O.M).....	249
What is an Operating Structure?.....	250

Two key facets.....	250
Where can the operational structure be found?.....	250
Creating a new operating structure.....	251
Defining taxonomy of the Organization.....	252
Configuring your operating structure.....	252
Defining Sample Organizational Chart.....	255
Organizational Chart.....	255
Business Validations for selection of level types.....	256
Users and Access groups.....	257
Utilization in CLM/KYC.....	257
Removal of Taxonomy Restrictions.....	258
Reusability of existing org charts to operating structure.....	258
Traditional methods of creating an Org chart (OOTB).....	258
How to reuse the existing Organizational Charts?.....	259
Converting Org chart to Operating structure.....	259
Importing Org units or divisions to Operating structure.....	259
Configuring data for product availability processing in PFFS.....	260
Current state of inclusion and exclusion of products.....	262
Organizational and product structures.....	262
Product matrix and inclusion or exclusion of cross-reference tables..	266
Methods of deriving available products.....	267
Class structure.....	268
Creating a sample financial product.....	269
Creating sample products using PDFS.....	270
Sample product structure.....	270
Product catalog ruleset.....	271
Attributes.....	272
Price.....	272
Eligibility.....	273
Benefits.....	274
Templates.....	275
Additional guidelines.....	279

Products	281
Bundles.....	281
Using an enrichment or screening marketplace component.....	281
Before you begin.....	282
Implementation in PFFS.....	284
Usecase examples.....	287
Implementing case types.....	288
Defining the case type modifications and attributes.....	288
PDFS	291
Product Designer for Financial Services installation guide.....	291
Completing the prerequisite tasks.....	291
Backing up your system.....	292
Installing the application.....	293
Product Designer for Financial Services update guide.....	298
Completing the prerequisite tasks.....	299
Backing up your system.....	299
Updating the application.....	300
Importing the application bundle.....	301
Reset Unique Ids.....	303
Unlock a new ruleset version.....	304
Optional: Updating sample data and operators.....	305
Optional: Enabling sample operator accounts.....	306
Product Designer for Financial Services implementation guide.....	306
Application overview.....	307
Initiate stage.....	309
Delivery stage.....	314
Production maintenance and monitoring.....	331
Technotes	333
Pega Foundation for Financial Services Accounting Guide.....	333
About this document.....	333
Intended audience.....	334
Accounts.....	334

Account constructs.....	334
Account setup and operation.....	335
Internal accounts.....	335
Chart of Accounts.....	336
Chart of Accounts lookup – how it works.....	336
Accounting party data.....	337
Payment types.....	338
Referenced by.....	338
Format.....	338
Accounting verification.....	340
Referenced by.....	341
Format.....	341
Accounting records affected.....	341
Accounting cutoff.....	342
Referenced by.....	343
Format.....	343
Accounting records affected.....	343
Cutoff operation.....	345
Hold cutoff accounting.....	346
Future cutoff accounting.....	346
Accounting rules.....	347
Adjustments.....	347
Suspense.....	348
Payables.....	348
Receivables.....	348
Outstanding balance.....	349
Suspense actions – Open, Raise, Lower, Close, and WriteOff.....	349
Configuring the accounting step rule.....	350
Rules affected.....	350
Accounting step rule fields.....	351
Naming conventions for accounting step rules.....	358
Using automatic processing flows and flow actions.....	358

Write-off.....	359
Small dollar write-off.....	360
How write off thresholds work.....	361
Aged write-off.....	362
Referenced by.....	363
Format.....	364
Accounting records affected.....	364
Duplicate and offset processing.....	366
Duplicate scoring rules.....	367
Item offset search.....	367
Financial offset search.....	368
Routing assignments to work queues.....	368
Verifying accounting functionality.....	369
Pega Foundation for Financial Services requirements guide.....	370
About this document.....	370
Intended audience.....	370
Overview.....	370
Using Requirement rules.....	371
Selecting Requirement objects.....	372
Document-based requirements.....	372
Document types.....	373
Configuring document entries in requirement rules.....	374
Unguided validation.....	376
Guided validation.....	377
Overall requirement validation.....	381
Flow-based requirements.....	386
Using Requirement set rules.....	389
Configuring requirement entries in requirement set rules.....	390
Selecting a requirement set.....	392
Configuring stage processing.....	395
Passing data between a requirement and its parent case.....	397
Passing scalar data using parameters.....	399

Passing data without parameters.....	402
Attachments in requirements.....	403
Attachment capabilities in Pega Platform.....	403
Attachments using CMIS.....	404
Configuration to support customer properties metadata.....	406
Technical details for the manage requirement documents screen.....	409
Configuring Pega Platform to act as a CMIS server.....	416
Requirement rule form help.....	421
About requirement rules.....	421
Rule resolution.....	423
Requirements - Completing the Requirement tab.....	423
Requirements - Completing the Parameters tab.....	427
Requirements - Completing the Subject Tags tab.....	429
Requirement set rule form help.....	429
About Requirement Set rules.....	430
Completing the New or Save As form.....	430
Rule resolution.....	431
Requirements - Completing the Requirement Rules tab.....	431
Requirements - Completing the Parameters tab.....	432
Pega Foundation for Financial Services requirements portal guide.....	434
About this document.....	434
Who should read this document?.....	434
Overview.....	435
Accessing the Requirements Portal.....	435
Understanding and building the structures in Requirements.....	435
Documents.....	436
Requirements.....	438
Requirement types.....	439
Guided document validation.....	439
Document group.....	440
Adding documents.....	440
Requirement configuration settings.....	443

Requirement-specific settings.....	444
Parameters.....	445
Requirement sets.....	446
Adding requirements.....	448
Setting parameters.....	449
Setting eForm preparation.....	450
Applicability and satisfaction logic.....	451
Simple example using the Requirements portal.....	452
Create document.....	452
Create requirement.....	454
Add a document to the requirement.....	455
Configure the document within the requirement.....	456
Add a parameter.....	457
Create a Requirement set.....	458
Review your created assets.....	461
Pega Foundation for Financial Services Version Verification Tech Note.....	461
Introduction.....	461
Verification.....	462
Resources	465

Get started

Explore the links below to access product documentation and other useful information for Pega Foundation for Financial Services

Learn about

- Release notes
- Product overview

Set up

- Install
- Update
- Implement

Technotes

- Accounting
- Requirement portal
- Version verification

PDFS Set up

- Install
- Update
- Implement

Additional

- Resources

Release notes

Pega Foundation for Financial Services provides several enhancements, known issues, and fixed issues in '23.

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

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

Pega Foundation for Financial Services '23 enhancements

Pega Foundation for Financial Services '23 includes the product enhancements that are described in this section.

- Extended support to deactivate data in reference tables while maintaining data integrity (Soft Delete)
- New integration with Equifax OneView API. Component is available in Pega Marketplace.
- Customer master profile is now enhanced with the following updates:
 - Audit tree modularization: Audit tree enhancements in customer master profile for performance improvement.
 - Master profile lite: Externalization of third party data from customer master profile for performance improvement. Customers have flexibility to enable this feature through DSS. A new dedicated table to store the third party data.
- Target Operating Model: Flex to choose the level type for the bottom most nodes (Branch) for the default and specialized taxonomies. Ability to automate the manual process of creation of Unit when WQs are created at Div level in Org chart.

Ability to automate the manual process of updation of the Unit value in WQ record when the divisions are imported from Org record which is created from OOTB LP.

Data structures and sample/reference data enhancements

- Data enhancements for Smart Dispute Model Workflow - ACH case type (Enabling Financial Institutions to process transactions in ACH network).
 - Addition of `T_OriginalTracenumber` in `fsf_sample_tran` table
 - Filter conditions updated from 'IsEqual' to 'contains' and marked 'ignore case' for RD `GetCreditCardTransactionSummary`, `DepositTransactionSummary`, `GetDepositTransactionSummary` for `MerchantDBA` column source.
- New columns created in `fsf_sample_tran` table to process transactions in ACH network for Smart Dispute Model workflow case type.
- New columns (CCT_TRANMODE, T_TRANMODE) created to hold 'mode of transaction' value in `fsf_sample_CCTran` and `fsf_sample_tran` tables.
- New column (acctnbr_text) added in `fsf_sample_authorization` table.
- Extended support to Alert and Investigation Management Accelerator (AIM) features through the following data model enhancements:
 - Added new columns (`fsf_sample_tran`, `fsf_sample_cctrans`, `fsf_sample_cmtrans`, `fsf_sample_loantran`) in transaction tables to hold.
 - recipient account info and originator account info
 - balance amount after every transaction
 - recipient country code
 - Added new column in `fsf_sample_bank` table to hold `BranchID` and `Branchname`
 - Updated `GetTransactionSummary` RD to include `TRANTYPEDESC`, `RecipientCountryCode`, `T_TRANTYPE` property.

- Updated `DepositAccountSummary` RD to include "CIFNBR" property.
- New DTs created to fetch abridged version of account summary –
`GetAccountSummaryMap`, `OpenSummaryMap`
- New DPs available to fetch the GICS codes from class *PegaFS-Data-FSF_REF_SUBINDUSTRY* (`D_GICSCodesOpen`, `D_GICSCodesValues`).

New classes and database changes

Pega Foundation for Financial Services '23 introduces several database changes to classes.

For the list of all database changes, go to: [Class and database changes](#), and [deprecated and withdrawn](#) rules in Pega Foundation for Financial Services.

UX enhancement to support Pega Constellation

React-based UI table sections that are available through OOTB platform configuration for improved usability (visibility, sorting, saving views, and so on) and experience.

Ruleset stack changes

There are no changes to ruleset stack in this release.

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.

There are no issues addressed in this release.



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

Known issues in this version

There are no *Known issues* 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 (INCs) in [My Support Portal](#). Ensure that you refer to the issue ID in all communications.

Updating from previous releases

Customers should consider the following information when updating from previous releases of Pega Foundation for Financial Services.

Note: For security purposes, the sample Pega operator accounts are disabled when you update to Pega Foundation for Financial Services '23. To continue to use any of these operators in your implementation, log in as an administrator and enable them. For more information, see the [Pega Foundation for Financial Services Update Guide](#) that is available on the [Pega Foundation for Financial Services product page](#) on Community.

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

Deprecated and withdrawn rules

Pega Foundation for Financial Services has deprecated some rules during this release to better maintain the product and make it easier for you to extend rules when needed. For more information, see [Pega Foundation for Financial Services '23 deprecated and withdrawn rules](#), and DB changes.

Removed classes

In the release of Pega Foundation for Financial Services '23, no classes were removed.

Deprecated and withdrawn rules

Pega Foundation for Financial Services 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 Foundation for Financial Services.

Release	Deprecated Rules	Withdrawn Rules
'23	75	12
8.8	4	4
8.7	24	50
8.6	121	15
8.5	54	10
8.4	9	19
8.3	28	14
8.2	68	143
8.1	98	22

Deprecated rules

This page lists the list of deprecated rules of Pega Foundation for Financial Services for 8.x releases.

'23 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for '23 release.

Name	Rule Type	Applies to (Class)	Ruleset
FinVerifyGetNextWork	Activity	Work-	PegaAccounting
FinStepVerifyVal	Activity	Work-	PegaAccounting
FinWorkBasketWriteOff	Activity	Work-	PegaFS

Name	Rule Type	Applies to (Class)	Ruleset
pyDefault	Data Transform	PegaFS-Data-DebitAccountDecisionResponses	PegaFS
pyMemo	Property	Link-	PegaFS
MortgageCreditReportSSN	Property	PegaReq-Data-ReqVerify-	PegaFSRequirements
pyButtonLabel Document Reviewed	Field Value	@baseclass	PegaFS
pyCaption Document ID	Field Value	@baseclass	PegaFS
pyCaption Role	Field Value	@baseclass	PegaFS
webwb pyCaseMgrPortalDualTabLeft_r180_clip.png	Binary File		PegaFSWeb
webwb pyCaseMgrPortalDualTabRight_r180.png	Binary File		PegaFSWeb
webwb pyCaseMgrPortalDualTabRight_r90.png	Binary File		PegaFSWeb
webwb pyCaseMgrPortalDualTabLeft_r90.png	Binary File		PegaFSWeb
webwb pyCaseMgrPortalDual	Binary File		PegaFSWeb

Name	Rule Type	Applies to (Class)	Ruleset
alTabLeft_r90_clip.png			
webwb pyCaseMgrPortalDu alTabRight_r90_clip png	Binary File		PegaFSWeb
ShowAuditDetails	Section	PegaFS-Work	PegaFS
ConstructPageHistory	Data Transform	PegaFS-Data-ChangeTracker	PegaFS
DisplayAuditForCustomer	Data Transform	Code-Pega-List	PegaFS
D_OpenDataAudit	Data Page	PegaFS-Data-ChangeTracker	PegaFS
D_ShowTargetElementPath	Data Page	PegaFS-Data-ChangeTracker	PegaFS
AuditHistoryDetails ForDeletedPages	Section	PegaFS-Data-ChangeTracker	PegaFS
AuditHistoryDetails ForDeletedPages	Flow Action	PegaFS-Data-ChangeTracker	PegaFS
RunAudit	Data Transform	PegaFS-Data-ChangeTracker	PegaFS
CountTaregetElementChanges	Report Definition	PegaFS-Data-ChangeTracker	PegaFS
GetLatestUpdateOf PageList	Report Definition	PegaFS-Data-ChangeTracker	PegaFS
TraverseBackPath	Data Transform	PegaFS-Work	PegaFS

Name	Rule Type	Applies to (Class)	Ruleset
BPMGUIAPIReloadT heHarness_back	Activity	@baseclass	PegaAccounting
CustomizedSaveAtt achment	Activity	Data-WorkAttach- Note	PegaFS
FinDocFilter	Activity	PegaAcct-Txn-	PegaAccounting
FinTxnFilter	Activity	PegaAcct-Txn-	PegaAccounting
Validate_New	Activity	Rule-PegaAcct- Financial- Adjustment	PegaAccounting
Validate	Activity	Rule-PegaAcct- Financial- PaymentType	PegaAccounting
FinActionLink	Activity	Work-	PegaAccounting
FinDisplayOffset	Activity	Work-	PegaAccounting
FinFollowupCorr	Activity	Work-	PegaAccounting
FinStepVerifSetWor kbasket	Activity	Work-	PegaAccounting
TC_D_AddedPagesI nfo	Test Case	PegaFS-Data- ChangeTracker	PegaFSFAUT
NewParty	Data Transform	Data-Party	PegaFS
pyActionPrompt Click to View Document	Field Value	@baseclass	PegaFSRequirements
pyCaption Customer Name	Field Value	@baseclass	PegaFS
pyCaption Documents	Field Value	@baseclass	PegaFS

Name	Rule Type	Applies to (Class)	Ruleset
pyCaption High	Field Value	@baseclass	PegaFS
NoDocs	Financial Payment Type	PegaAcct-Txn-	PegaFSRequirements
webwb pyCaseMgrPortalDualTabLeft_r270.png	Binary File		PegaFSWeb
webwb pyCaseMgrPortalDualTabLeft_r180.png	Binary File		PegaFSWeb
webwb pyCaseMgrPortalDualTabRight_r180_clip.png	Binary File		PegaFSWeb
webwb pyCaseMgrPortalDualTabRight_r270.png	Binary File		PegaFSWeb
UpdateCardAccount	Flow Action	PegaFS-Data-Account-Card	PegaFS
RemoveSummaryColumns	Data Transform	Code-Pega-List	PegaFS
ResponseAuditDetails	Data Transform	Code-Pega-List	PegaFS
RunAuditDetailsForCustomer	Data Transform	Code-Pega-List	PegaFS
SetPathForAuditDetails	Data Transform	Code-Pega-List	PegaFS

Name	Rule Type	Applies to (Class)	Ruleset
GetPageAdditionInfo	Data Transform	PegaFS-Data-ChangeTracker	PegaFS
D_AddedPagesInfo	Data Page	PegaFS-Data-ChangeTracker	PegaFS
D_TargetPath	Data Page	PegaFS-Data-ChangeTracker	PegaFS
GetChangeTrackerDetailsForCustomer	Report Definition	PegaFS-Data-ChangeTracker	PegaFS
GetPageAddedDetailsAudit	Report Definition	PegaFS-Data-ChangeTracker	PegaFS
IsSourceForPageListLevel	When	PegaFS-Data-ChangeTracker	PegaFS
RunAuditDetails	Data Transform	PegaFS-Work	PegaFS
ShowWorkObjectList	Activity	Code-Pega-List	PegaFS
SortAccountList	Activity	PegaAcct-Party-InternalAccount	PegaAccounting
FinStepVerifSetWorkBasket	Activity	PegaAcct-Step-	PegaAccounting
AddWorkParties	Flow Action	PegaFS-Work	PegaFS
CustomerBackgroundVerification	Flow Action	PegaFS-Work	PegaFS
AppDisplayOffset	Activity	PegaAcct-Work	PegaFS
AppHistorySearchResult	Activity	PegaAcct-WorkAttach-FinOffsets	PegaAccounting

Name	Rule Type	Applies to (Class)	Ruleset
AppHistorySearchResult	Activity	PegaAcct-WorkAttach-NonFinOffsets	PegaAccounting
ShowSingleWorkObject	Activity	PegaFS-Work	PegaFS
copyrules	Activity	Rule-PegaAcct-Financial-Adjustment	PegaFS
Validate	Activity	Rule-PegaAcct-Financial-Adjustment	PegaAccounting
copyrules	Activity	Rule-PegaAcct-Financial-Payable	PegaAccounting
FinBatchWriteOff	Activity	Work-	PegaFS
FinGetCOSPrinters	Activity	Work-	PegaFS
pyCaption Select	Field Value	@baseclass	PegaFS
pyCountryCode SRB	Field Value	@baseclass	PegaFS

8.8 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.8 release.

Name	Label	Rule Type	Applies to (Class)	Ruleset version
RelCodeLocalizedDescription	RelCodeLocalizedDescription	Property	PegaFS-Data-RelCodes	08-08-01

Name	Label	Rule Type	Applies to (Class)	Ruleset version
AsyncProcessingConfig	Async processing config	Decision Table	System-Queue-DefaultEntry	08-08-01
AsyncProcessingConfig	Async processing config	Data Transform	@baseclass	08-08-01

8.7 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.7 release.

Rule Name	Label	Object Class	Class Name
ListDepositAccounts	List deposit accounts section	Rule-HTML-Section	PegaFS-Landing
ListLoanTransactions	List Loan Transactions Section	Rule-HTML-Section	PegaFS-Landing
FSIFTrackSecurityChangesLogic_ext	FSIFTrackSecurityChangesLogic_ext	Rule-Obj-Activity	Embed-ModelParams
ListCMAccounts	List CM Account records	Rule-HTML-Section	PegaFS-Landing
FSFSampleDataManagerMain	Sample data management section	Rule-HTML-Section	PegaFS-Landing
DisplayIlliquidAssets	Display illiquid assets	Rule-HTML-Section	PegaFS-Landing

Rule Name	Label	Object Class	Class Name
FSFSampleDataInne r	Sample data management tabs section	Rule-HTML-Section	PegaFS-Landing
ListLiabilities	List Liabilities	Rule-HTML-Section	PegaFS-Landing
FSFSampleDataMan agementOuter	Fsf sample data management outer	Rule-HTML-Section	PegaFS-Landing
ListHouseholdRecor ds	List household records section	Rule-HTML-Section	PegaFS-Landing
pyCaption Sample Data Management	Sample data management	Rule-Obj-FieldValue	@baseclass
DisplayLiquidAssets	Display liquid assets	Rule-HTML-Section	PegaFS-Landing
ListDepositTransacti ons	List deposit transactions section	Rule-HTML-Section	PegaFS-Landing
FSFSampleDataMan agement	Fsf sample data management	Rule-Obj-FlowAction	PegaFS-Landing
ListCustomers	List customers	Rule-HTML-Section	PegaFS-Landing
ListCCTransactions	List Credit Card Transactions	Rule-HTML-Section	PegaFS-Landing
FSFSampleDataMan agementInner	Fsf sample data management inner	Rule-HTML-Section	PegaFS-Landing
FSFSampleDataMan agement	Fsf sample data management	Rule-HTML-Section	PegaFS-Landing
ListInvestmentTrans actions	List Investment Transactions	Rule-HTML-Section	PegaFS-Landing
ListOfficerRecords	List Officer Record	Rule-HTML-Section	PegaFS-Landing
ListCardAccounts	List Card Accounts	Rule-HTML-Section	PegaFS-Landing

Rule Name	Label	Object Class	Class Name
FSFSampleDataManagementInner_Header	Fsf sample data management inner header panel	Rule-HTML-Section	PegaFS-Landing
ListCCMerchants	List Credit Card Transactions	Rule-HTML-Section	PegaFS-Landing
ListInvestmentPositionRecords	List Investment Position Records Section	Rule-HTML-Section	PegaFS-Landing

8.6 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.6 release.

Rule Name	Label	Object Class	Class Name
AccountTypeIndicator	Display icons according to type of account	Rule-HTML-Property	
CloneLoanTransactions	Clone loan transactions	Rule-Obj-Model	PegaFS-Data-AccountXRef
CloneDepositTransactions	Cloned deposit transactions	Rule-Obj-Model	PegaFS-Data-AccountXRef
CloneInvestmentTransactions	Clone investment transactions	Rule-Obj-Model	PegaFS-Data-AccountXRef
CloneCardTransactions	Clone card transactions	Rule-Obj-Model	PegaFS-Data-Account-
YesNoFees	Drop box for use with yes no properties	Rule-HTML-Property	

Rule Name	Label	Object Class	Class Name
CloneCommercialName	Clone commercial name	Rule-Obj-Property	PegaFS-Landing
SelectWorkParty	Select work party	Rule-HTML-Property	
CloneLoanAccount	Clone loan account	Rule-Obj-Model	PegaFS-Data-AccountXRef
DisableCloneButton	Disable clone button	Rule-Obj-When	PegaFS-Landing
CloneAccounts	Clone accounts	Rule-Obj-Model	PegaFS-Data-AccountXRef
PerformDefaults	BPM Extension Point API:Customizes Perform functionality	Rule-Obj-Activity	PegaFS-Work-Goal
CloneInvestmentAccount	Clone investment account	Rule-Obj-Model	PegaFS-Data-AccountXRef
ReadOnlyYesNo	Read only yes no	Rule-HTML-Property	
FSFSampleDataManagerMain CU	FSFSampleDataManagerMain CU	Rule-HTML-Harness	PegaFS-Landing
MyBUFavorites	My BU Favorites	Rule-Obj-Report-Definition	PegaFS-Data-Favorite
IsCMPPParameterizationApplicable	Is CMP Parameterization applicable	Rule-Obj-When	PegaFS-Work
MyHHFavorites	My H H Favorites	Rule-Obj-Report-Definition	PegaFS-Data-Favorite

Rule Name	Label	Object Class	Class Name
CloneDepositAccount	Clone deposit account	Rule-Obj-Model	PegaFS-Data-AccountXRef
CloneAccountCardXRefDetails	Clone account card xref details	Rule-Obj-Model	PegaFS-Data-AccountXRef
pyCaption CopyMessage	CopyMessage	Rule-Obj-FieldValue	PegaFS-Landing
CloneAccountCardXRefDetails	Clone account card xref details	Rule-Obj-Model	PegaFS-Data-Account-
MyClientFavorites	My H H Favorites	Rule-Obj-Report-Definition	PegaFS-Data-Favorite
CloneCardAccount	Clone card account	Rule-Obj-Model	PegaFS-Data-AccountXRef
ServiceLevelBrokenItems	Service level broken items	Rule-Obj-Report-Definition	System-Queue-ServiceLevel
CloneParty2IdDetails	Clone party 2 id details	Rule-Obj-Model	PegaFS-Landing
ReadOnlyYesNoForNumeric	Read only yes no for numeric	Rule-HTML-Property	
CloneLoanAccount	Clone loan account	Rule-Obj-Property	PegaFS-Landing
RelevantPartyDecision	Relevant party decision	Rule-Declare-DecisionTable	PegaFS-Data-PartyPartyXRef
CopySuccess	CopySuccess	Rule-HTML-Paragraph	PegaFS-Landing
CustomRadioButton	Custom radio button	Rule-HTML-Property	
CloneParty1IdDetails	CloneParty1IdDetails	Rule-Obj-Model	PegaFS-Landing

Rule Name	Label	Object Class	Class Name
SetCopyFlag	Set copy flag	Rule-Obj-Model	PegaFS-Landing
CloneCustomer	Clone customer	Rule-Obj-Model	PegaFS-Landing
FlowDependencyBrokenItems	Flow dependency broken items	Rule-Obj-Report-Definition	System-Queue-FlowDependency
CloneCardAccount	Clone card account	Rule-Obj-Property	PegaFS-Landing
ClonePartyDetails	ClonePartyDetails	Rule-Obj-Model	PegaFS-Landing
CloneDepositAccount	Clone deposit account	Rule-Obj-Property	PegaFS-Landing
DuplicateCases	Duplicate cases	Rule-HTML-Property	
CloneAccountDetails	Clone account details	Rule-Obj-Model	PegaFS-Landing
ClonedAccountNumbers	Cloned account numbers	Rule-Obj-Property	PegaFS-Landing
CloneCustomerDetails	Clone customer details	Rule-Obj-Model	PegaFS-Landing
DisableCloneButtonOrg	Disable clone button org	Rule-Obj-When	PegaFS-Landing
CloneCustomerCommsDetails	Clone customer comms details	Rule-Obj-Model	PegaFS-Landing
CloneCustomerName	Clone customer name	Rule-Obj-Property	PegaFS-Landing
CloneInvestmentAccount	Clone investment account	Rule-Obj-Property	PegaFS-Landing
GetAllowedProductsByLocation	Get Allowed Products By Location	Rule-Obj-Activity	Code-Pega-List

Rule Name	Label	Object Class	Class Name
WorkListGadget	Work list gadget	Rule-HTML-Section	Data-Portal
AccountLevelAuthorization	Account level authorization	Rule-HTML-Section	PegaFS-Data-Account-
AddNewAccountSection	Add new account section	Rule-HTML-Section	PegaFS-Data-Account-
SmartInfoForAcctNbr	Smart info for acct nbr	Rule-HTML-Section	PegaFS-Data-Account-Card
CustomerAccounts	Customer accounts	Rule-HTML-Section	PegaFS-Data-Account-
AddDepositAccount	New account details	Rule-HTML-Section	PegaFS-Data-Account-Deposit
AuthContactDetails	Auth contact details	Rule-HTML-Section	PegaFS-Data-AccountXRef
AuthorizedOwner	Authorized owner	Rule-HTML-Section	PegaFS-Data-AccountXRef
DisplaySelectedAcctNumbers	Select products for contact text	Rule-HTML-Section	PegaFS-Data-AccountXRef
DisplayProductNameWrapper	Display product name wrapper	Rule-HTML-Section	PegaFS-Data-AcctProdXRef
ServiceContacts	Service contacts	Rule-HTML-Section	PegaFS-Data-Account-
TransactionsHeader	Transactions header	Rule-HTML-Section	PegaFS-Data-Account-
SummaryPageContainerSection	Summary page container section	Rule-HTML-Section	PegaFS-Data-Party
AccountDetailsWrapper	Account details wrapper	Rule-HTML-Section	PegaFS-Data-AccountXRef

Rule Name	Label	Object Class	Class Name
CustomerDocuments	Customer documents	Rule-HTML-Section	PegaFS-Data-Party
SelectProductsForAccount	Select products for account	Rule-HTML-Section	PegaFS-Data-AccountXRef
AccountDetails	Customer accounts	Rule-HTML-Section	PegaFS-Data-Account-
AuthContacts	Auth contacts	Rule-HTML-Section	PegaFS-Data-Account-
AuthContactsServices	Auth contacts and services	Rule-HTML-Section	PegaFS-Data-Account-
AutoPaymentSmartInfo	Auto payment smart info	Rule-HTML-Section	PegaFS-Data-Account-
NewAccountDetails	New account details	Rule-HTML-Section	PegaFS-Data-Account-Deposit
ViewAccountPrimaryDetails	View account primary details	Rule-HTML-Section	PegaFS-Data-Account-Deposit
OtherAddressDetails	Other address details	Rule-HTML-Section	PegaFS-Data-Account-
DisplayAccountDetails	Display customer accounts details	Rule-HTML-Section	PegaFS-Data-Account-
DisplayProducts	Display products	Rule-HTML-Section	PegaFS-Data-Account-
AddCMAccount	Add cm account	Rule-HTML-Section	PegaFS-Data-Account-Investment
AssociateCapMarAccount	Associate cap market account to	Rule-HTML-Section	PegaFS-Data-Account-Investment

Rule Name	Label	Object Class	Class Name
HouseholdAccountDetails	House hold account details	Rule-HTML-Section	PegaFS-Data-Account-Investment
LinkedRecommendations	Linked recommendations	Rule-HTML-Section	PegaFS-Data-Account-
OtherAddressDetailsSection	Other address details section	Rule-HTML-Section	PegaFS-Data-Account-
RecommendationDetailInfo	Recommendation detail info	Rule-HTML-Section	PegaFS-Data-Account-
ShowAutoPayMessage	Show auto pay message	Rule-HTML-Section	PegaFS-Data-Account-
AuthorizedCIFNumber	Authorized cif number	Rule-HTML-Section	PegaFS-Data-AccountXRef
AuthContactDelete	Auth contact delete	Rule-HTML-Section	PegaFS-Data-AcctCardXref
CardTransactions	Card transactions	Rule-HTML-Section	PegaFS-Data-AcctCardXref
AddNewAccount	Add New Account	Rule-HTML-Section	PegaFS-Data-Account-Deposit
AssociateDepositAccount	Associate account to	Rule-HTML-Section	PegaFS-Data-Account-Deposit
DissociateCustomerAccount	Dissociate customer account	Rule-HTML-Section	PegaFS-Data-Account-
SelectUsersForAccountList	Select users for account list	Rule-HTML-Section	PegaFS-Data-Account-
SetAutoPay	Set auto pay	Rule-HTML-Section	PegaFS-Data-Account-

Rule Name	Label	Object Class	Class Name
ViewAccountPrimaryDetails	View account primary details	Rule-HTML-Section	PegaFS-Data-Account-
DisplaySelectedAccountNumbers	Select products for contact text	Rule-HTML-Section	PegaFS-Data-AccountXRef
CloneButton	Clone button	Rule-HTML-Section	PegaFS-Landing
SelectUsersForAccount	Select users for account	Rule-HTML-Section	PegaFS-Data-Account-
AuthContactDelete	Auth contact delete	Rule-HTML-Section	PegaFS-Data-AccountXRef
ExistingCustomer	Existing customer	Rule-HTML-Section	PegaFS-Work-NewAccount
CustomerHeaderInfo	Header section for work object	Rule-HTML-Section	PegaFS-Work
pyWorkBodyRight	Right column of the perform and review harnesses	Rule-HTML-Section	PegaFS-Work
pyWorkParties	Display repeating list of parties in this work object	Rule-HTML-Section	PegaFS-Work
pyWorkPartiesWrapper	py work parties wrapper	Rule-HTML-Section	PegaFS-Work
pyWorkRelatedDetails	Mini tabs on the right of perform or review harnesses	Rule-HTML-Section	PegaFS-Work
AddLoanAccount	New loan account details	Rule-HTML-Section	PegaFS-Data-Account-Loan
SelectProductsForAccount	Select products for account	Rule-HTML-Section	PegaFS-Data-Account-

Rule Name	Label	Object Class	Class Name
CopyUtilityForCommercialBanking	Copy utility for commercial banking	Rule-HTML-Section	PegaFS-Landing
CopyUtilityForRetailBanking	Copy utility for retail banking	Rule-HTML-Section	PegaFS-Landing
DisplayClonedResults	Display Cloned Results	Rule-HTML-Section	PegaFS-Landing
FSFSampleCopyUtility	Sample data copy utility form	Rule-HTML-Section	PegaFS-Landing
FSFSampleDataInne_r_CU	Sample Data Management tabs section	Rule-HTML-Section	PegaFS-Landing
EmploymentDetails	Employment Details	Rule-HTML-Section	PegaFS-Work
Extreme	Extreme	Rule-HTML-Section	PegaFS-Data-RiskProfile
CustomerBackgroundVerification	Customer Background Verification	Rule-HTML-Section	PegaFS-Work
CaptureIndividualPartyDetails	Capture individual party details	Rule-HTML-Section	PegaFS-Data-Party-Ind
AddCardAccount	Add card account	Rule-HTML-Section	PegaFS-Data-Account-Card
DisplayAccountDetails	Display account details	Rule-HTML-Section	PegaFS-Data-Party
AccountLevelAuthorizationList	Account level authorization list	Rule-HTML-Section	PegaFS-Data-Account-
ProductDetails	Customer accounts	Rule-HTML-Section	PegaFS-Data-Account-

Rule Name	Label	Object Class	Class Name
SelectProducts	Select products for accounts	Rule-HTML-Section	PegaFS-Data-Account-
CustomerAddressDetails	Legal address details	Rule-HTML-Section	PegaFS-Data-Account-
AuthorizedContactNumber	Authorized contact number	Rule-HTML-Section	PegaFS-Data-AccountXRef
IconInsert	Icon insert	Rule-HTML-Property	
RecentWorkListGadget	Recent work list gadget	Rule-HTML-Property	
ViewLWODocAttachments	View attachments	Rule-HTML-Property	
RecentWorkList	Recent work list	Rule-HTML-Section	Data-Portal
ReqDocSection	Req doc section	Rule-HTML-Section	PegaReq-Data-ReqDocEntry

8.5 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.5 release.

Rule Name	Label	Object Class	Class Name
ModHyperlink	Mod hyperlink overridden by mycofsreq	Rule-HTML-Property	
LocalActionLink	Local action link	Rule-HTML-Property	

Rule Name	Label	Object Class	Class Name
D_TriggerBloomberg_DocumentsBatch	Trigger enrichment documents (batch) using Bloomberg	Rule-Declare-Pages	PegaFS-Data-ExtProvider-BusService-eEnrichment
DisplayModalLocalAction	Display modal local action	Rule-HTML-Property	
findWorkByIDQueryNext	Find existing Work by ID or query, or get next	Rule-Obj-HTML	Data-Gadget
FilterHighestVersion	Filter highest version	Rule-Obj-When	Rule-PegaAcct-
ValidateReqDoc	Validate req doc	Rule-HTML-Property	
OverlayHTML	OverlayHTML	Rule-Obj-HTML	@baseclass
FinPickOrigParty	Pick from all possible parties	Rule-HTML-Property	
Google_link	Google link	Rule-HTML-Property	
D_TriggerClarient_EntityDetails	Trigger enrichment using Clarient	Rule-Declare-Pages	PegaFS-Data-ExtProvider-BusService-eEnrichment
WorkItemIDNewWindow	Work item id new window	Rule-HTML-Property	
BloombergEntityDataMapping	Bloomberg entity data mapping	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment

Rule Name	Label	Object Class	Class Name
IsBloomberg	Is bloomberg	Rule-Obj-When	PegaFS-Data-ExtProvider-BusService-
ResetTempClarientProperties	Reset temp clarient properties	Rule-Obj-Model	PegaFS-Data-ExtProvider-Clarient
AppOpenPotential	script function to open potentialitem	Rule-HTML-Fragment	
GetDocumentByID_Bloomberg	GetDocumentByID_Bloomberg	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
RelatedItemID	Related item id	Rule-HTML-Property	
DS_OperatorListSuspiciousActivity	Ds operator list suspicious activity	Rule-HTML-Property	
FSIF_RV_Eventhandlers	Rv eventhandlers	Rule-HTML-Fragment	
MapBloombergDummmyDocResponse	MapBloombergDummmyDocResponse	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
TriggerBloomberg_SearchEntity	TriggerBloomberg_SearchEntity	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
SetAutomaticBloombergId	Set automatic bloomberg id	Rule-Obj-Model	PegaFS-Data-ExtProvider-

Rule Name	Label	Object Class	Class Name
			BusService-eEnrichment
DocumentsBatchBl oomberg	DocumentsBatchBl oomberg	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
ProductID	Product identifier	Rule-Obj-Property	PegaFS-Data-Product
SetPageClass	Set page class	Rule-Obj-Model	@baseclass
PegaOrgDivUnitStru cture	PegaOrgDivUnitStru cture	Rule-Obj-When	Data-Admin-
D_TriggerBloomber g_GetDocumentByI D	Trigger enrichment documents (docID) using Bloomberg	Rule-Declare-Pages	PegaFS-Data-ExtProvider-BusService-eEnrichment
DeleteDocument	Delete document	Rule-HTML-Property	
D_TriggerClarient_S earchEntity	Trigger enrichment search using Clarient	Rule-Declare-Pages	PegaFS-Data-ExtProvider-BusService-eEnrichment
ClarientAttachAllDo cuments	Clarient Attach All Documents	Rule-Obj-Activity	PegaFS-Data-ExtProvider-Clarient
IsClarientCompon entPresent	Is clarient component present	Rule-Obj-When	@baseclass
AppDisplaySearchR esults	Display Attached Search Results	Rule-Obj-HTML	PegaAcct-WorkAttach-SearchResults

Rule Name	Label	Object Class	Class Name
PegaBankOrgStructure	Pega bank org structure	Rule-Obj-When	Data-Admin-
TriggerBloomberg_EntityDetails	Trigger bloomberg entity details	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
SmartResolveItemID	Smart resolve item id	Rule-HTML-Property	
TriggerBloomberg_EntityDocuments	TriggerBloomberg_EntityDocuments	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
D_TriggerBloomberg_EntityDocuments	Trigger enrichment documents (entityID) using Bloomberg	Rule-Declare-Pages	PegaFS-Data-ExtProvider-BusService-eEnrichment
D_HomePurchaseType	Home purchase types	Rule-Declare-Pages	Code-Pega-List
BloombergEntityDummmyResponse	BloombergEntityDummmyResponse	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
DateShiftLoanTransactionTest	Applies date shift to loan transaction records	Rule-Obj-Activity	PegaFS-Int-FSF_SAMPLE_LOANTRAN
FinGetVerificationWork	Links to manage group's work	Rule-Obj-HTML	Data-Gadget

Rule Name	Label	Object Class	Class Name
D_TriggerBloomberg_EntityDetails	Trigger enrichment using Bloomberg	Rule-Declare-Pages	PegaFS-Data-ExtProvider-BusService-eEnrichment
TypesOfHomes	Types of homes	Rule-Obj-Model	Code-Pega-List
CaptureReqDetails	Capture requirement details	Rule-Obj-HTML	Rule-
BloombergDummy Response	BloombergDummy Response	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
RefreshWList	Refresh wlist	Rule-HTML-Property	
OpenContactType	Open contact type	Rule-HTML-Property	
WorkManagerFS	FS Foundation Work Manager	Rule-Portal	
myGroupDD	Links to manage group's work in Designer Desktop	Rule-Obj-HTML	Data-Gadget
TriggerClarient_EntityDetails	TriggerClarient_EntityDetails	Rule-Obj-Model	PegaFS-Data-ExtProvider-BusService-eEnrichment
AppSearchCriteriaDisplay	Display of Criteria used in searching	Rule-Obj-HTML	PegaAcct-WorkAttach-SearchResults

Rule Name	Label	Object Class	Class Name
WorkInACover	Work in a cover	Rule-HTML-Property	
EditDocument	Edit document	Rule-HTML-Property	

8.4 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.4 release.

Rule name	Label	Rule type	Class
D_RelProductList	List of products in accounts	Rule-Declare-Pages	PegaFS-Data-Product
PreSelectUsersForAccount	Pre select users for account	Rule-Obj-Activity	PegaFS-Data-Account-
OpenFSF_SAMPLE_CCACCT	Open the Account Details from FSF_SAMPLE_CCACC T Table	Rule-Obj-Activity	PegaFS-Int-FSF_SAMPLE_CCACCT
BuildOrganizationDetails	BuildOrganizationDetails	Rule-Obj-Activity	@baseclass
PostAddCustomer	PostAddCustomer	Rule-Obj-Activity	PegaFS-Work-CustomerSearch
GetLoanAccountDetails	Get Loan AccountDetails	Rule-Obj-Activity	PegaFS-Data-AccountXRef
ResolveReviewMatch	ResolveReviewMatch	Rule-Obj-Model	Data-
BaseAmount	BaseAmount	Rule-Obj-Property	PegaFS-Data-Transaction-

Rule name	Label	Rule type	Class
Account	Account ID	Rule-Obj-Property	PegaFS-Data-Beneficiary

8.3 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.3 release.

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
AppRateLooku p	Activity	@baseclass	PegaAccountin g	08-03-01
AppClearWork PagePotential	Activity	PegaAcct- WorkAttach- Duplicates	PegaAccountin g	08-03-01
AppSearchDup icates	Activity	Work-	PegaAccountin g	08-03-01
FinWorkBasket WriteOff	Activity	Work-	PegaAccountin g	08-03-01
WorkUserFS	Portal		PegaFS	08-03-01
SetupRecordCr eationContext	Activity	@baseclass	PegaFS	08-03-01
OpenXRef	Activity	PegaFS-Data- AccountXRef	PegaFS	08-03-01
OpenBlobObje ct	Activity	@baseclass	PegaFS	08-03-01
PreDisplayFiel ds	Activity	@baseclass	PegaFS	08-03-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
PostGetTableNameLinkedToAppClass	Activity	Code-Pega-List	PegaFS	08-03-01
DeleteOrgLayer	Activity	Data-Admin-	PegaFS	08-03-01
PrePopulatedChildLevel	Activity	Data-Admin- OrgLevel	PegaFS	08-03-01
UpdateOrgStructure	Activity	Data-Admin- OrgLevel	PegaFS	08-03-01
PopulatePartyListWithName	Activity	Data-Party	PegaFS	08-03-01
GetMktSegLookupValues	Activity	PegaFS-Data- Account-	PegaFS	08-03-01
CheckDuplicateTypes	Activity	PegaFS-Data- Party	PegaFS	08-03-01
GetAllWorkParties	Activity	PegaFS-Work	PegaFS	08-03-01
AddressType Previous Address	Field Value	PegaFS-Data- Party-Ind	PegaFS	08-03-01
SetObjClass	Data Transform	@baseclass	PegaFS	08-03-01
OpenMap	Data Transform	PegaFS-Data- IlliquidAssets	PegaFS	08-03-01
SaveMap	Data Transform	PegaFS-Data- IlliquidAssets	PegaFS	08-03-01
FSUser	Portal		PegaFS	08-03-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
OpenFSF_SAM PLE_LU_MARK ETSEGMENT	Activity	PegaFS-Int- FSF_REF_LU_M ARKETSEGMENT	PegaFSInt	08-03-01
OpenFSF_SAM PLE_CCACCT	Activity	PegaFS-Int- FSF_SAMPLE_C CACCT	PegaFSInt	08-03-01
OpenFSF_SAM PLE_CUST	Activity	PegaFS-Int- FSF_SAMPLE_C UST	PegaFSInt	08-03-01
OpenFSF_SAM PLE_CUSTACCT XREF	Activity	PegaFS-Int- FSF_SAMPLE_C USTACCTXREF	PegaFSInt	08-03-01
UpdateFSF_SA MPLE_INVEST MENTMARKET DATA	Activity	PegaFS-Int- FSF_SAMPLE_I NVESTMENTM KTDATA	PegaFSInt	08-03-01
SaveFSAttach ments	Activity	@baseclass	PegaRequirements	08-03-01

8.2 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.2 release.

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
Linked contacts	Property	PegaFS-Data- Product	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
LOB	Property	PegaFS-Data-Product	PegaFS	08-02-01
Node4Value	Property	PegaFS-Data-Product	PegaFS	08-02-01
ProductCategory	Property	PegaFS-Data-Product	PegaFS	08-02-01
ProductCode	Property	PegaFS-Data-Product	PegaFS	08-02-01
Product description	Property	PegaFS-Data-Product	PegaFS	08-02-01
ProductID	Property	PegaFS-Data-Product	PegaFS	08-02-01
ProductName	Property	PegaFS-Data-Product	PegaFS	08-02-01
ProductType	Property	PegaFS-Data-Product	PegaFS	08-02-01
RelAccountList	Property	PegaFS-Data-Product	PegaFS	08-02-01
RelUserList	Property	PegaFS-Data-Product	PegaFS	08-02-01
RowSelected	Property	PegaFS-Data-Product	PegaFS	08-02-01
SKUID	Property	PegaFS-Data-Product	PegaFS	08-02-01
Product description	Property	PegaFS-Data-ProductType	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
ProductType	Property	PegaFS-Data-ProductType	PegaFS	08-02-01
Declare_ProductTypePage	Data Page	PegaFS-Data-ProductType	PegaFS	08-02-01
D_CustAcctProducts	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
D_Product	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
D_ProductOpen	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
D_ProductSelect	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
D_ProductsForAccount	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
D_ProductsList	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
D_ProductSummary	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
D_ProductType	Data Page	PegaFS-Data-ProductType	PegaFS	08-02-01
D_RelProductList	Data Page	PegaFS-Data-Product	PegaFS	08-02-01
CustomerDetailsForCLM	Section	PegaFS-Work	PegaFS	08-02-01
LoadMasterProfile	Activity	PegaFS-Data-Party-MasterProfile	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
LoadMasterProfile	Activity	PegaFS-Data-Party-MasterProfile	PegaFS	08-02-02
UpdateAvailableProductStatus	Activity	PegaFS-Data-ProductMatrix	PegaFS	08-02-01
pyCaptionAuthorizedUsers	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
pyCaptionFNMAConforming 30 Year Fixed Rate Mortgage	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Basic Checking	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName FNMA Conforming 30 Year Fixed Rate Mortgage	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Gold Credit Card	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Home Equity Loan	Field Value	PegaFS-Data-Product	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
ProductName Personal Gold Savings Account	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Premium Business Checking	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Premium Checking	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Student Loan	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Traditional IRA	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductName Vehicle Loan	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
pyCaption AddOneOrMoreAccounts	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
pyCaption Student Loan	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
pyCaption Vehicle Loan	Field Value	PegaFS-Data-Product	PegaFS	08-02-01
ProductDescription Individual	Field Value	PegaFS-Data-ProductType	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
Retirement Savings Accounts				
ProductDescription Personal Checking Accounts	Field Value	PegaFS-Data- ProductType	PegaFS	08-02-01
ProductDescription Personal Credit Card Accounts	Field Value	PegaFS-Data- ProductType	PegaFS	08-02-01
ProductDescription Personal Loans	Field Value	PegaFS-Data- ProductType	PegaFS	08-02-01
ProductDescription Personal Saving Accounts	Field Value	PegaFS-Data- ProductType	PegaFS	08-02-01
ProductDescription Small Business Checking Account	Field Value	PegaFS-Data- ProductType	PegaFS	08-02-01
CopyNecessaryProductDetails	Data Transform	PegaFS-Data- Product	PegaFS	08-02-01
OnDeselectOfProduct	Data Transform	PegaFS-Data- Product	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
OnSelectOfProduct	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
OpenMap	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
OpenMapProductAdoption	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
PreAddProduct	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
PreUpdateProduct	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
ProductOpenMap	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
pyDefault	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
SaveMap	Data Transform	PegaFS-Data-Product	PegaFS	08-02-01
pyDefault	Data Transform	PegaFS-Data-ProductType	PegaFS	08-02-01
DataTableEditorReport	Report Definition	PegaFS-Data-ProductType	PegaFS	08-02-01
LoadProductLists	Report Definition	PegaFS-Data-ProductType	PegaFS	08-02-01
GetProduct	When	PegaFS-Data-Product	PegaFS	08-02-01
IsAssociated	When	PegaFS-Data-Product	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
IsProductRequi resKYC	When	PegaFS-Data-Product	PegaFS	08-02-01
IsUserCifNbrNotNull	When	PegaFS-Data-Product	PegaFS	08-02-01
LinkedCustomers	Property	PegaFS-Data-Product	PegaFS	08-02-01

8.1 Deprecated rules

The following table lists the deprecated rules of Pega Foundation for Financial Services for 8.1 release.

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
AppRateLooku p	Activity	@baseclass	PegaAccountin g	08-01-01
BPMGUIAPIRel oadTheHarnes s_back	Activity	@baseclass	PegaAccountin g	08-01-01
SetComponent Label	Activity	Embed- PegaAcct- Search- Scoring-List- Comp	PegaAccountin g	08-01-01
SortAccountList	Activity	PegaAcct- Party- InternalAccoun t	PegaAccountin g	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
FinStepVerifSet WorkBasket	Activity	PegaAcct-Step-	PegaAccountin g	08-01-01
FinDocFilter	Activity	PegaAcct-Txn-	PegaAccountin g	08-01-01
FinTxnFilter	Activity	PegaAcct-Txn-	PegaAccountin g	08-01-01
AppHistorySea rchResult	Activity	PegaAcct- WorkAttach- FinOffsets	PegaAccountin g	08-01-01
AppHistorySea rchResult	Activity	PegaAcct- WorkAttach- NonFinOffsets	PegaAccountin g	08-01-01
FinFollowupCo rr	Activity	Work-	PegaAccountin g	08-01-01
FinBatchWrite Off	Activity	Work-	PegaAccountin g	08-01-01
Validate	Activity	Rule-PegaAcct- Financial- Adjustment	PegaAccountin g	08-01-01
Validate_New	Activity	Rule-PegaAcct- Financial- Adjustment	PegaAccountin g	08-01-01
copyrules	Activity	Rule-PegaAcct- Financial- Payable	PegaAccountin g	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
Validate	Activity	Rule-PegaAcct-Financial-PaymentType	PegaAccounting	08-01-01
AppSearchDuplicates	Activity	Work-	PegaAccounting	08-01-01
FinActionLink	Activity	Work-	PegaAccounting	08-01-01
FinDisplayOffset	Activity	Work-	PegaAccounting	08-01-01
FinStepVerifSetWorkbasket	Activity	Work-	PegaAccounting	08-01-01
FinStepVerifVal	Activity	Work-	PegaAccounting	08-01-01
FinVerifGetNextWork	Activity	Work-	PegaAccounting	08-01-01
DeletePartyPartyXRef	Activity	PegaFS-Data-PartyPartyXRef	PegaFS	08-01-01
DeleteAcctRelCodes	Activity	PegaFS-Data-AcctRelCodes	PegaFS	08-01-01
ClearNewAddressMessage	Activity	PegaFS-Data-Address	PegaFS	08-01-01
DeleteLastAddress	Activity	PegaFS-Data-Address	PegaFS	08-01-01
InsertErrorMessageIntoViewer	Activity	@baseclass	PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
IterateCurrencyList	Activity	@baseclass	PegaFS	08-01-01
MarkItAttachAllDocuments	Activity	@baseclass	PegaFS	08-01-01
AppendBlankContact	Activity	Code-Pega-List	PegaFS	08-01-01
AppendBlankMember	Activity	Code-Pega-List	PegaFS	08-01-01
ShowWorkObjectList	Activity	Code-Pega-List	PegaFS	08-01-01
TestSplitAndJoin	Activity	Code-Pega-List	PegaFS	08-01-01
TestAddWorkGroup	Activity	Data-Admin-	PegaFS	08-01-01
ValidateOneRuleSet	Activity	Data-Rule-Summary	PegaFS	08-01-01
CustomizedSaveAttachment	Activity	Data-WorkAttachment-Note	PegaFS	08-01-01
RuleHistoryByOperator	Activity	History-Rule	PegaFS	08-01-01
RefineSearchResults	Activity	Index-Customer	PegaFS	08-01-01
FSUpdateOrgUnitWrapper	Activity	Pega-Landing-Org	PegaFS	08-01-01
AppDisplayOffset	Activity	PegaAcct-Work	PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
ClearOtherSelectionsforAutho rize	Activity	PegaFS-Data- Account-	PegaFS	08-01-01
ClearOtherSele ctionsInAccoun t	Activity	PegaFS-Data- Account-	PegaFS	08-01-01
GetCustomerA ccountDetails	Activity	PegaFS-Data- Account-	PegaFS	08-01-01
ListCreditCard AcctCustomers	Activity	PegaFS-Data- AccountXRef	PegaFS	08-01-01
ListLoanAcctCu stomers	Activity	PegaFS-Data- AccountXRef	PegaFS	08-01-01
PreAddBenefic iary	Activity	PegaFS-Data- Beneficiary	PegaFS	08-01-01
GetLei	Activity	PegaFS-Data- Party-Org	PegaFS	08-01-01
BusinessCount ryListDelete	Activity	PegaFS-Data- Country	PegaFS	08-01-01
GetNewBusine ssCountry	Activity	PegaFS-Data- Country	PegaFS	08-01-01
ListHousehold Members	Activity	PegaFS-Data- Household	PegaFS	08-01-01
UpdateHouseh oldRecord	Activity	PegaFS-Data- Household	PegaFS	08-01-01
AddInvestmen tPositionRecor d	Activity	PegaFS-Data- InvestmentPos ition	PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
DeleteCustomerDataForSyncAbandon	Activity	PegaFS-Data-Party	PegaFS	08-01-01
EditRelationshipsForHouseHold	Activity	PegaFS-Data-Party	PegaFS	08-01-01
ListCustOpenCases	Activity	PegaFS-Data-Party	PegaFS	08-01-01
RetrievePrimConInformation	Activity	PegaFS-Data-Party	PegaFS	08-01-01
RetrieveCreditHistory	Activity	PegaFS-Data-Party-Ind	PegaFS	08-01-01
AddNewMerchant	Activity	PegaFS-Data-Party-Org-Merchant	PegaFS	08-01-01
PreAddNewMerchant	Activity	PegaFS-Data-Party-Org-Merchant	PegaFS	08-01-01
DeleteRelatedPartiesForSyncAbandon	Activity	PegaFS-Data-PartyPartyXRef	PegaFS	08-01-01
RelCodesCRUD	Activity	PegaFS-Data-RelCodes	PegaFS	08-01-01
PreAddDepositTransaction	Activity	PegaFS-Data-Transaction-Deposit	PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
PreAddInvestmentTransaction	Activity	PegaFS-Data-Transaction-Investment	PegaFS	08-01-01
ShowWorkObjectList	Activity	PegaFS-Data-VisualizerCollection	PegaFS	08-01-01
RestorePartyTypeSampleData	Activity	PegaFS-Landing	PegaFS	08-01-01
AccountLookup	Activity	PegaFS-Work	PegaFS	08-01-01
BuildUnifiedDocumentCheckList	Activity	PegaFS-Work	PegaFS	08-01-01
DiscardInformationTransactions	Activity	PegaFS-Work	PegaFS	08-01-01
fsAddWorkObjectParty	Activity	PegaFS-Work	PegaFS	08-01-01
GetCIFAndCustomerInfo	Activity	PegaFS-Work	PegaFS	08-01-01
GetCustAcctXref	Activity	PegaFS-Work	PegaFS	08-01-01
IsCustomerSelected	Activity	PegaFS-Work	PegaFS	08-01-01
PostAccountsFlow	Activity	PegaFS-Work	PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
PostAddProducts	Activity	PegaFS-Work	PegaFS	08-01-01
RemovePreviousAssociations	Activity	PegaFS-Work	PegaFS	08-01-01
ValidateRefineByFields	Activity	PegaFS-Work	PegaFS	08-01-01
SearchCommercialCustomers	Activity	PegaFS-Work	PegaFS	08-01-01
ShowSingleWorkObject	Activity	PegaFS-Work	PegaFS	08-01-01
TestMasterProfile	Activity	PegaFS-Work	PegaFS	08-01-01
TestWorkAndMasterProfile	Activity	PegaFS-Work	PegaFS	08-01-01
AssociateHouseholdMember	Activity	PegaFS-Work-CustomerSearch	PegaFS	08-01-01
LinkCustomerAccount	Activity	PegaFS-Work-NewAccount	PegaFS	08-01-01
CopyLocalizableRulesToHigherVersion	Activity	Rule-	PegaFS	08-01-01
Harve_CopyLocalizableRulesToHigherVersion	Activity	Rule-	PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
FinBatchWriteOff	Activity	Work-	PegaFS	08-01-01
ReportOfRules	Activity	Rule-	PegaFS	08-01-01
copyrules	Activity	Rule-PegaAcct-Financial-Adjustment	PegaFS	08-01-01
CustomerLookUp	Activity	Work-	PegaFS	08-01-01
FedWireOutbound	Activity	Work-	PegaFS	08-01-01
FinGetCOSPrinters	Activity	Work-	PegaFS	08-01-01
FinWorkBasketWriteOff	Activity	Work-	PegaFS	08-01-01
GetCIFAndCustomerInfo	Activity	Work-	PegaFS	08-01-01
GetTaxIDProperties	Activity	Work-	PegaFS	08-01-01
SaveCustomNotesAttachmentsWrapper	Activity	Work-	PegaFS	08-01-01
List	List	PegaFS-Data-LinkedIDVerificationDocs	PegaFS	08-01-01
List	List	PegaFS-Data-Product	PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
List	List	PegaFS-Data-ProductType	PegaFS	08-01-01
Testact	Activity	PegaReq-Work-ProcessRequirement	PegaFSRequirements	08-01-01
PersonalRuleCheckOut	Activity	Rule-	PegaRequirements	08-01-01

Withdrawn rules

This page lists the list of withdrawn rules of Pega Foundation for Financial Services for 8.x releases.

'23 Withdrawn rules

Name	Rule Type	Applies to (Class)	Ruleset
CheckNbr	Property	PegaFS-Data-Transaction-Deposit	PegaFS
CheckNbr	Property	PegaFS-Data-Transaction-Loan	PegaFS
webwb labeledmarker.js	Text File		PegaFS
webwb markermanager.js	Text File		PegaFS
pyCaption Deutsche Bundesbank	Field Value	@baseclass	PegaFS
pyCaption Type	Field Value	@baseclass	PegaFS

Name	Rule Type	Applies to (Class)	Ruleset
pyCaption Deutsche Borse	Field Value	@baseclass	PegaFS
Exchange Deutsche	Field Value	PegaFS-Data-InvestmentSecurity	PegaFS
IsCMPParameterizationApplicable	When	PegaFS-Work	PegaFS
pyNewWorkPartyDetail	Section	Work-	PegaFSRequirements
FSFSampleDataManagerMainMenu	Navigation	Pega-Landing	PegaPDFSFoundation
pyStatusWork Open	Field Value	Work-	PegaRequirements

8.8 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.8 release.

Name	Label	Rule Type	Class Name	Ruleset version
IsRetailCustomer	IsRetailCustomer	When	PegaFS-Data-Party	08-08-01
pyCaption DB	Db	Field Value	@baseclass	08-08-01
CustomerType Organization	Taxid type for organizations	Field Value	@baseclass	08-08-01
CustomerType Individual	Taxid type for individuals	Field Value	@baseclass	08-08-01

8.7 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.7 release.

Rule Name	Object Class	Class Name	Ruleset
pyShowMyCases	Rule-HTML-Section	Data-Portal	PegaFS
pyWorkBasketAssgnCount	Rule-HTML-Section	Data-Admin-WorkBasket	PegaFS
evidenceSource	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant-Entity	PegaFS
evidenceEffectiveDate	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant-Entity	PegaFS
country	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant	PegaFS
pyDashboardDetails	Rule-HTML-Section	Data-Admin-	PegaFS
pyUserWorkList	Rule-HTML-Section	Data-Admin-	PegaFS
IsClariantReportsExist	Rule-Obj-When	PegaFS-Data-ExtProvider-Clariant	PegaFS
pyDefaults	Rule-Obj-Activity	Data-UIGallery-Features-Extension	PegaFS
FSUpdateAddChildPage	Rule-Obj-Activity	Data-Admin-OrgDivision	PegaFS
ReasonNotToProvideTIN Account Holder is unable to obtain it	Rule-Obj-FieldValue	PegaFS-Data-TaxResidence	PegaFS

Rule Name	Object Class	Class Name	Ruleset
access	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent	PegaFS
SelectedItem	Rule-HTML-Section	PegaFS-Data-ExtProvider-Clarent	PegaFS
ResetTempClarentProperties	Rule-Obj-Model	PegaFS-Data-ExtProvider-Clarent	PegaFS
AVID	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent	PegaFS
pyMiniPartyDisplayDetailShow	Rule-Obj-FlowAction	Data-Party	PegaFS
clarentApprovedInd	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent-Entity	PegaFS
canViewEntity	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent	PegaFS
pyPega7ExpressPortalContent	Rule-HTML-Section	Data-Portal-PegaExpress	PegaFS
pyRegion	Rule-Obj-Property	PegaFS-Data-Party-MasterProfile	PegaFS
ClarentAttachAllDocuments	Rule-Obj-Activity	PegaFS-Data-ExtProvider-Clarent	PegaFS
ClarentAttachAllDocuments_duplicatecheck	Rule-Obj-Activity	PegaFS-Data-ExtProvider-Clarent	PegaFS
ClarentEntityReports	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent	PegaFS

Rule Name	Object Class	Class Name	Ruleset
documentTypeCD	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant-Entity	PegaFS
pyTagDashboardHeader	Rule-HTML-Section	Data-Social-Tag	PegaFS
pyExcludePegaRulesets	Rule-Obj-When	Embed-Guardrails	PegaFSRequirements
documentId	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant-Entity	PegaFS
FSUpdateAddChildPage	Rule-Obj-Activity	Data-Admin-	PegaFS
isRevoked	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant	PegaFS
Selclariantentitydetails	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant	PegaFS
MapAllProps	Rule-Obj-Activity	PegaFS-Int-FSF_SAMPLE_CUSTA CCTXREF	PegaFSInt
entityLegalName	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant	PegaFS
evidenceDetails	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant	PegaFS
externalSourceType	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant-Entity	PegaFS
sourceType	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant	PegaFS

Rule Name	Object Class	Class Name	Ruleset
FilesAttached	Rule-Obj-When	PegaFS-Data-ExtProvider-Clarent	PegaFS
pyCMMMyWorklist	Rule-HTML-Section	Data-Portal	PegaFS
pyWorkBasketDetail	Rule-HTML-Section	Data-Admin-WorkBasket	PegaFS
ClarentEntityReport s	Rule-HTML-Section	PegaFS-Data-ExtProvider-Clarent	PegaFS
FSIFUserDashboard	Rule-HTML-Section	@baseclass	PegaFS
pyWorkGroupBaske ts	Rule-HTML-Section	Data-Portal	PegaFS
documentTypeDesc	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent-Entity	PegaFS
GetGICSCodes	Rule-Obj-Report-Definition	PegaFS-Int-FSF_REF_SUBINDUS TRY	PegaFSRequirements
pyDefault	Rule-Obj-Model	Data-Portal	PegaFS
UpdateAddChildPageWrapper	Rule-Obj-Activity	Pega-Landing	PegaFS
pageNumber	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent-Entity	PegaFS
evidenceCreationDate	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clarent-Entity	PegaFS
FSOrgChartRightClickMenu	Rule-Navigation	Data-Admin-OrgUnit	PegaFS

Rule Name	Object Class	Class Name	Ruleset
pyTeamMembersWidget	Rule-HTML-Section	Data-Portal	PegaFS
evidenceField	Rule-Obj-Property	PegaFS-Data-ExtProvider-Clariant-Entity	PegaFS

8.6 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.6 release.

Rule Name	Label	Object Class	Class Name
Google_link	Google link	Rule-HTML-Property	
SaveMap	Save map	Rule-Obj-Model	PegaFS-Int-FSF_PARTYRELATTRIBXREF
providerAVID	Provider AVID	Rule-Obj-Property	PegaFS-Int-ExtProvider-Clariant-SearchLEOILeResults
ShowWorkObjectList	displays all objects on code pega list page, for now	Rule-Obj-Activity	PegaFS-Data-VisualizerCollection
OpenMap	Open map	Rule-Obj-Model	PegaFS-Int-FSF_PARTYRELATTRIBXREF
Perform	Perform work object	Rule-HTML-Harness	Work-

Rule Name	Label	Object Class	Class Name
pyPopulateCaseContentsWrapper	Populate case contents wrapper	Rule-Obj-Activity	Work-
DisplayDocuments2G	DisplayDocuments2G	Rule-HTML-Section	@baseclass
pyMiniPartyDetails	Generic display of party details in a small gadget	Rule-HTML-Section	PegaFS-Data-Party
RNGDocumentsSummary	Display documents summary	Rule-HTML-Section	PegaFS-Work
DisplayRequirementCases	DisplayRequirementCases	Rule-HTML-Section	PegaFS-Work
ShowRiskDetail	ShowRiskDetail`	Rule-HTML-Section	PegaFS-Work
RNGDisplayPendingReqSummary	Display Pending Requirements	Rule-HTML-Section	PegaFS-Work
RNGDisplayRequirementList	Display Requirement List	Rule-HTML-Section	PegaFS-Work-Requirement
RNGViewAllRequirements	Display All Requirements	Rule-HTML-Section	PegaFS-Work

8.5 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.5 release.

Rule Name	Label	Object Class	Class Name
pyWorkBasketAssignmentsDefinition	Outstanding WorkBasket Assignments Report	Rule-Obj-Report-Definition	Assign-WorkBasket

Rule Name	Label	Object Class	Class Name
PegaFFS_Schema 08.04	Pega Foundation for Financial Services 8.4 Schema Layer	Rule-Admin-Product	
pyWorklistAssignme ntsDefinition	Outstanding Worklist Assignments Report	Rule-Obj-Report- Definition	Assign-Worklist
PegaFFS_Int 08.04	Pega Foundation for Financial Services 8.4 Integration Layer	Rule-Admin-Product	
AddressType Previous Address	Previous Address	Rule-Obj-FieldValue	PegaFS-Data-Party- Ind
PegaFFS_SnapDatal nstances 08.04	Pega Foundation for Financial Services 8.4 Snap Data Instances	Rule-Admin-Product	
ResolveReviewMatc h	ResolveReviewMatc h	Rule-Obj-Model	Data-
PegaFFS_DataInstan ces 08.04	Pega Foundation for Financial Services 8.4 Data Instances	Rule-Admin-Product	
PegaFFS_SnapSche ma 08.04	Pega Foundation for Financial Services 8.4 Schema Layer	Rule-Admin-Product	
UpgradeOnOpen	Upgrade on open	Rule-Obj-Activity	PegaFS-Int- FSF_RELCODES

8.4 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.4 release.

Rule name	Label	Rule type	Class
CountryNameSpeci alTranslation	CountryNameSpeci alTranslation	Rule-Declare- DecisionTable	PegaFS-DATA- Country
IndexWorkBasketIn OrgChart	IndexWorkBasketIn OrgChart	Rule-Declare-Index	Data-Admin-
D_GetRequirement SetList	RequirementSets	Rule-Declare-Pages	Rule- RequirementSet
pyUploadFile	pyUploadFile	Rule-Obj-Activity	Data-WorkAttach- File
DeleteDataInstanc es	Delete Data Instances	Rule-Obj-Activity	@baseclass
LA_RISKSCORE	LA_RISKSCORE	Rule-Obj-Property	PegaFS-Int- FSF_SAMPLE_LOAN ACCT
BankRoutingNumbe rAutoPay	BankRoutingNumbe rAutoPay	Rule-Obj-Property	PegaFS-Data- AutoPayment
AccountBalanceWor k	AccountBalanceWor k	Rule-Obj-Property	PegaFS-Work
EIN	EIN	Rule-Obj-Property	PegaFS-Data- Branch-
CCT_BASEAMOUNT	CCT_BASEAMOUNT	Rule-Obj-Property	PegaFS-Int- FSF_SAMPLE_CCTR A
BankAccountNumb erAutoPay	BankAccountNumb erAutoPay	Rule-Obj-Property	PegaFS-Data- AutoPayment

Rule name	Label	Rule type	Class
TodaysChangePercent	TodaysChangePercent	Rule-Obj-Property	PegaFS-Data-InvestmentSecurity
AccountNumber	Account identifier which the transaction posts against	Rule-Obj-Property	PegaFS-Data-Transaction-
T_BASEAMOUNT	T_BASEAMOUNT	Rule-Obj-Property	PegaFS-Int-FSF_SAMPLE_TRAN
TodaysChangePercent	TodaysChangePercent	Rule-Obj-Property	PegaFS-Data-Account-Investment
AccountNumber	AccountNumber	Rule-Obj-Property	PegaFS-Data-AcctProdXRef
Balance	Balance	Rule-Obj-Property	PegaFS-Data-Asset-IlliquidAsset
AccountNumber	Account number	Rule-Obj-Property	PegaFS-Data-Account-
AccountNumber	AccountNumber	Rule-Obj-Property	PegaFS-Work

8.3 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.3 release.

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
IndexWorkGroupInOrgChart	Declare Index	Data-Admin-	PegaFS	08-03-01
.SubIndustryRiskCode	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-03-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
GetTotalResultCount	Activity	@baseclass	PegaFS	08-03-01
pyModalPageCancelAction	Activity	@baseclass	PegaFS	08-03-01
SetMarketSegment	Data Transform	PegaFS-Data-Account-	PegaFS	08-03-01
EvaluateAMLCDDProfileForLinkedRelatedPartyList	Data Transform	PegaFS-Work	PegaFS	08-03-01
FetchDivisionsSubsidiariesAndOwner	Report Definition	PegaFS-Int-FSF_PARTYPARTYXREF	PegaFS	08-03-01
GetAllowedProductsByLOB	Report Definition	PegaFS-Int-FSF_PRODUCTMATRIX	PegaFS	08-03-01
RetrieveCustomerName	Report Definition	PegaFS-Int-FSF_SAMPLE_CUST	PegaFS	08-03-01
AC_ACCTNBR	Property	PegaFS-Int-FSF_SAMPLE_ACCTCARDXREF	PegaFSInt	08-03-01
PrimaryEmailPhoneFax	Section	PegaFS-Data-Party	PegaFSInt	08-03-01
GetMktSegLookupValues	Activity	PegaFS-Data-Account-	PegaFSInt	08-03-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
AcctMarketSegmentOpenMap	Data Transform	PegaFS-Data-Account-	PegaFSInt	08-03-01
GetAttachmentReference	Activity	Work-	PegaRequirements	08-03-01

8.2 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.2 release.

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
pyMiniPartyDetails	Section	Data-Party-Person	PegaAccounting	08-02-01
pyCaseAssets	Section	PegaAcct-Work	PegaAccounting	08-02-01
pyCaseAssetsParticipants	Section	PegaAcct-Work	PegaAccounting	08-02-01
pyCaseBody	Section	PegaAcct-Work	PegaAccounting	08-02-01
pyNewCaseContainerWithWarning	Section	PegaAcct-Work	PegaAccounting	08-02-01
AssetAcctNbr	Property	PegaFS-Data-IlliquidAssets	PegaFS	08-02-01
AssetAddressCity	Property	PegaFS-Data-IlliquidAssets	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
AssetAddressCountry	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetAddressKey	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetAddressLine1	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetAddressLine2	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetAddressLine3	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetAddressLine4	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetAddressZipCode	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetDescription	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetId	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetReleased	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
AssetType	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
CurrentValue	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
CurrValueDate	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
Lca_colldescription	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
Lca_collid	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
Lca_colrtype	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
LoanAccountNumber	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
Rank	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
VLOCLimit	Property	PegaFS-Data- IlliquidAssets	PegaFS	08-02-01
PP_ISDEEMED CONTROLLING	Property	PegaFS-Int- FSF_PARTYPAR TYXREF	PegaFS	08-02-01
.SubIndustryName	Declare Expression	PegaFS-Data- RiskProfile	PegaFS	08-02-01
.SubIndustryRiskCode	Declare Expression	PegaFS-Data- RiskProfile	PegaFS	08-02-01
.SubIndustryRiskScore	Declare Expression	PegaFS-Data- RiskProfile	PegaFS	08-02-01
.YearsOfWork	Declare Expression	PegaFS-Data- RiskProfile	PegaFS	08-02-01
.SubIndustryID	Declare Expression	PegaFS-Data- RiskProfile	PegaFS	08-02-01
.VulnerableValue	Declare Expression	PegaFS-Data- ProductMatrix	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
.BusinessCode	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.BusinessCode Description	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.BusinessCode RiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.BusinessCode Score	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.BusinessCode Sensitivity	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.BusinessCountriesScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.BVulnerableBusinessCode	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryOfBirth	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryOfBirthScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryOfIncorporation	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryOfIncorporationScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryOfNationality	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryOfNationalityScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
.CountryOfResidence	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryOfResidenceScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryRiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CountryRiskScoreValue	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CustomerRiskCode	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.CustomerRiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.DurationOfRelationshipRiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.DurationOfRelationshipScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.eCredit	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.eFraud	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.eID	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.eOFAC	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.eScreening	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
.ExternalDataRiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.ExternalDataRiskValue	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.OccupationRiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.OccupationRiskValue	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.PositionCode	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.PositionCodeScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.ProductVulnerabilitiesScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.ProductVulnerabilityRiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.RelatedPartyRiskScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.RelevantRelationshipScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
.SubIndustryCodeScore	Declare Expression	PegaFS-Data-RiskProfile	PegaFS	08-02-01
pyWorkListWidgetGridsAuto	Section	Data-Portal	PegaFS	08-02-01
pyDisplayStages	Section	Work-	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
CreateRequirementObject	Activity	PegaReq-Data-ProcessRequirement	PegaFS	08-02-01
CustomerTypeFund	Field Value	@baseclass	PegaFS	08-02-01
LoadMasterProfileCommitTime	Data Transform	PegaFS-Data-Party-MasterProfile	PegaFS	08-02-02
GetRelationDetails	Report Definition	PegaFS-Int-FSF_PARTYPARTYXREF	PegaFS	08-02-01
PartyPartyXRefGetChildrenWithDetails	Report Definition	PegaFS-Int-FSF_PARTYPARTYXREF	PegaFS	08-02-01
PartyPartyXRefGetParents	Report Definition	PegaFS-Int-FSF_PARTYPARTYXREF	PegaFS	08-02-01
CheckCustomerID	When	PegaFS-Data-PartyPartyXRef	PegaFS	08-02-01
PerfectIncome	Requirements	PegaFS-Data-Income	PegaFS	08-02-01
MortgageApplication	Requirements	PegaFS-Work-NewAccount	PegaFS	08-02-01
MortgagePersonalIdentification	Requirements	PegaFS-Work-NewAccount	PegaFS	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
MortgageUnde rwritingReqSet	Requirements ets	PegaFS-Work- NewAccount	PegaFS	08-02-01
D_CollateralEv aluationOpen	Data Page	PegaFS-Data- CollateralEvalu ation	PegaFS	08-02-01
CR_EVALBY	Property	PegaFS-Int- FSF_SAMPLE_C OLLATERALEVA LUATION	PegaFSInt	08-02-01
CR_EVALSOUR CE	Property	PegaFS-Int- FSF_SAMPLE_C OLLATERALEVA LUATION	PegaFSInt	08-02-01
CR_EVALSOUR CEDATE	Property	PegaFS-Int- FSF_SAMPLE_C OLLATERALEVA LUATION	PegaFSInt	08-02-01
CR_EVALVALUE	Property	PegaFS-Int- FSF_SAMPLE_C OLLATERALEVA LUATION	PegaFSInt	08-02-01
IA_MILAGE	Property	PegaFS-Int- FSF_SAMPLE_IL LIQUIDASSET	PegaFSInt	08-02-01
IA_NOOFBATH ROOMS	Property	PegaFS-Int- FSF_SAMPLE_IL LIQUIDASSET	PegaFSInt	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
IA_NOOFBEDR OOMS	Property	PegaFS-Int- FSF_SAMPLE_IL LIQUIDASSET	PegaFSInt	08-02-01
IA_LASTPREMI UMAMOUNTP AID	Property	PegaFS-Int- FSF_SAMPLE_IL LIQUIDASSET	PegaFSInt	08-02-01
IA_LASTPREMI UMAMPAIDDA TE	Property	PegaFS-Int- FSF_SAMPLE_IL LIQUIDASSET	PegaFSInt	08-02-01
pyActionArea_ ScreenFlow	Section	PegaFS-Work	PegaFSRequirements	08-02-01
W2	Requirements	PegaFS-Work- NewAccount	PegaFSRequirements	08-02-01
pxCMISEnable d	When	@baseclass	PegaFSRequirements	08-02-01
Personal Loan	Requirements	PegaFS-Work- NewAccount	PegaFSRequirements	08-02-01
Vehicle Loan	Requirements	PegaFS-Work- NewAccount	PegaFSRequirements	08-02-01
IdentityProof	Requirements	PegaReq-Data- ReqVerify-	PegaFSRequirements	08-02-01
Income	Requirements	PegaReq-Data- ReqVerify-	PegaFSRequirements	08-02-01
Mortgage Application	Requirements	PegaReq-Data- ReqVerify-	PegaFSRequirements	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
MortgageClosingAgentContract	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
MortgageCreditReport	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
MortgageHUD_1	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
MortgageIncome	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
MortgagePersonalIdentification	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
Personal Loan	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
ProofOfResidence	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
Residence Proof	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
Vehicle Loan	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
W2	Requirements	PegaReq-Data-ReqVerify-	PegaFSRequirements	08-02-01
IncomeReqSet	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01
LoansReqSet	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
MortgageClosingReqSet	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01
MortgageProcessingReqSet	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01
MortgageUnderwritingReqSet	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01
Sample Manual Add Requirement Set	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01
Work Related Req Set	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01
Work Status Req Set	RequirementSets	PegaFS-Work-NewAccount	PegaFSRequirements	08-02-01
HomeLoan	RequirementSets	PegaReq-Work-ProcessRequirement	PegaFSRequirements	08-02-01
MortgageClosingAgentContact	RequirementSets	PegaReq-Work-ProcessRequirement	PegaFSRequirements	08-02-01
pyReportEditor Header	Section	Code-Pega-List	PegaPDFSFoundation	08-02-01
pyTeamMembersWidgetRow	Section	Data-Admin-Operator-ID	PegaPDFSFoundation	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
SaveFSAttach ments	Activity	@baseclass	PegaRequirem ents	08-02-01
SaveFSFDocAtt achments	Activity	@baseclass	PegaRequirem ents	08-02-01
pyUseDevelop erSkin	When	Rule-	PegaRequirem ents	08-02-01
Driver Licence	Requirements	PegaReq-Data- ProcessRequir ement	PegaRequirem ents	08-02-01
Bank Agreement	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01
BusinessAddre ss	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01
BusinessLicens e	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01
Citizenship	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01
Company Constitution Document	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01
CompanyName	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01
GreenCard	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01
IndividualAddr essDomicile	Requirements	PegaReq-Data- ReqVerify-	PegaRequirem ents	08-02-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
IndividualBackgroundCheck	Requirements	PegaReq-Data-ReqVerify-	PegaRequirements	08-02-01
Individual Identity	Requirements	PegaReq-Data-ReqVerify-	PegaRequirements	08-02-01
Individual Nationality	Requirements	PegaReq-Data-ReqVerify-	PegaRequirements	08-02-01
Individual Non Political Exposure	Requirements	PegaReq-Data-ReqVerify-	PegaRequirements	08-02-01
RegisteredAddress	Requirements	PegaReq-Data-ReqVerify-	PegaRequirements	08-02-01
State ID	Requirements	PegaReq-Work	PegaRequirements	08-02-01
Mortgage Loan	Requirements	PegaReq-Work-ProcessRequirement	PegaRequirements	08-02-01
MortgageLoan Set	Requirements	PegaReq-Work-ProcessRequirement	PegaRequirements	08-02-01

8.1 Withdrawn rules

The following table lists the withdrawn rules of Pega Foundation for Financial Services for 8.1 release.

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
pyAttachmentFeedDescription	Control		CMISPlus	08-01-01
AccountingSkin	Skin		PegaAccounting	08-01-01
ECMNodePath	Property	Work-	PegaFS	08-01-01
pyCaptionDocuments	Field Value	PegaFS-Data-Party	PegaFS	08-01-01
AddressTypeHeadquarters	Field Value	PegaFS-Data-Party-Ind	PegaFS	08-01-01
ContactType Previous Employer 1	Field Value	PegaFS-Data-Party-Ind	PegaFS	08-01-01
ContactType Previous Employer 2	Field Value	PegaFS-Data-Party-Ind	PegaFS	08-01-01
IsRMInChargeRequiredAtBusRelList	When	PegaFS-Data-BusRelationship	PegaFS	08-01-01
EndUser62_NB_BCD	Skin		PegaFS	08-01-01
FSIF	Skin		PegaFS	08-01-01
CompositeBrand_FSIF	Skin		PegaFS	08-01-01
EndUser62	Skin		PegaFS	08-01-01
FSApplicationSkin	Skin		PegaFS	08-01-01

Name	Rule Type	Applies to (Class)	Ruleset	Ruleset version
PegaFinancialS ervicesBrand	Skin		PegaFS	08-01-01
SampleDataCo nfigurationPan el_FSIF	Skin		PegaFS	08-01-01
pyDashboard7	Harness	Data-Portal	PegaPDFSFoun dation	08-01-01
pyOverlayTem plate	Section	@baseclass	PegaPDFSFoun dation	08-01-01
pyRecents	Section	Data-Portal	PegaPDFSFoun dation	08-01-01
pyOverlayTem plate	Section	@baseclass	PegaRequirem ents	08-01-01
SaveFSAttach ment	Activity	Data- WorkAttach- ProcessRequir ements	PegaRequirem ents	08-01-01
SaveFSFDocAtt achment	Activity	Data- WorkAttach- ProcessRequir ements	PegaRequirem ents	08-01-01
DisplayAttach ment	HTML	@baseclass	PegaRequirem ents	08-01-01

Install

- **Install**

Install

- **Backing up your system**
- **Completing the prerequisite tasks**

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
- pegarules.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.

Completing the prerequisite tasks

Before you install Pega Foundation for Financial Services, complete the following tasks.

1. Install or update to Pega Platform '23 or the latest patch release.

For more information, see the *Pega Platform Installation Guide* or the *Pega Platform Update Guide* for your environment on the [Pega Platform page on Pega Community](#).

Patch releases are available through Pega's standard software delivery process.

You can request software through Pega Software Distribution or by contacting GCS.

2. Update the online help to the latest version from the Online Help Files section on the Pega Platform Update page. After you download the latest prhelp.war file, deploy it in your installation before continuing. For deployment information, see the [Pega Platform Installation Guide for your application server](#).

3. Follow the update path instructions as mentioned below:

Updating from release 7.17 or 7.21: Updating from older versions of Pega Foundation for Financial Services is a multi-step process, therefore updating directly from release 7.17 to 8.23 is not recommended due to schema updates in releases 7.21 and 7.21.1. Follow these steps for the suggested best approach:

- Upgrade from release 7.17 to 7.21
- Upgrade from release 7.21 to 7.21.1
- Upgrade from release 7.21.1 to 8.23

For more information, see the *Pega Foundation for Financial Services Update Guide* for a previous release on the [Pega Foundation for Financial Services product page](#) before proceeding further.

Updating from 7.21.1, 7.22, 7.31, 7.32, 7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8: You can update to release 8.23 directly from 7.21.1, 7.22, 7.31, 7.32, 7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, or 8.8. If you are updating to Pega Foundation for Financial

Services '23 from release 7.21.1, 7.22, 7.31, 7.32, 7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, or 8.8 and plan to update the sample data after updating the application, drop the sample and snapshot tables before importing the application.

4. Review the database policies and application permissions that are used by your instance of Pega Platform to determine whether the application is permitted to update the database automatically, or if you must generate the database scripts that your organization will use make schema changes manually.

Installing the application

To install Pega Foundation for Financial Services, complete the following tasks.

Before you begin:

Make sure that you back up your system and that you complete all the prerequisite tasks. For more information, see [Backing up your system](#) and [Completing the prerequisite tasks](#).

Importing the application

Use the Import wizard to import the system data and rulesets for Pega Foundation for Financial Services to Pega Platform.

1. Log in to Pega Platform at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.

For example:

`administrator@pega.com`

2. In the header of Dev Studio, click Configure > Application > Distribution > Import and perform the following actions:

- a. Depending on your web browser, click Browse or Choose File, select the \Rules\PegaFFS_Int.jar file from your distribution image, and then click Next.
- b. Ensure that you import all components by leaving the Enable advanced mode to provide granular control over the import process check box cleared, and then click Next.

Result:

The wizard identifies differences between the database schema of your current system and the schema of Pega Foundation for Financial Services '23.

- c. If the file that you are importing requires changes to the database schema, select Automatic or Manual, and then click Next.
If you select Manual, see [Viewing and applying schema changes](#).
 - d. Follow the instructions on the screen, and when the import is completed, click Done.
3. Repeat step 2 for the following files:
 - \Rules\PegaFFS.jar
 - \Rules\PegaFFS_Sample.jar
 - \Rules\PegaFFS_Schema.jar
 - \Rules\PegaFFS_DataInstances.jar
 4. Apply the required hotfixes by using Hotfix Manager.
For more information, see [Pega Foundation for Financial Services Hotfixes](#).
 5. **Optional:** Install the Pega Product Designer for Financial Services.
For more information, see the *Pega Product Designer for Financial Services Installation Guide* that is available on the [Pega Foundation for Financial Services product page](#).
 6. **Optional:** For the best performance and user experience, pre-assemble the rules in the application by running the Static Assembler utility.

For more information, see [Preassembling rules in an application by using the Static Assembler utility](#).

7. In the **Explorer** panel, click Records > SysAdmin > RuleSet, and then verify that the *PegaFS* ruleset appears on the list.

Optional: Importing sample data

Pega Foundation for Financial Services includes some sample customer data that you can use to explore the application's functionality.

1. Log in to at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.

For example:

`administrator@pega.com`

2. In the header of Dev Studio, click Configure > Application > Distribution > Import and perform the following actions:
 - a. Depending on your web browser, click Browse or Choose File, select the `\SampleData\PegaFFS_SampleSchema.jar` from your distribution image, and then click Next.
 - b. Ensure that you import all components by leaving the **Enable advanced mode to provide granular control over the import process** check box cleared, and then click Next.

Result:

The wizard identifies differences between the database schema of your current system and the schema of Pega Foundation for Financial Services '23.

- c. If the file that you are importing requires changes to the database schema, select Automatic or Manual, and then click Next.

- If you select **Manual**, see [Viewing and applying schema changes](#).
- d. Follow the instructions on the screen, and when the import is completed, click **Done**.
 3. Repeat step 2 for the following files:
 - \SampleData\PegaFFS_SampleDataInstances.jar
 - \SampleData\PegaFFS_SnapSchema.jar
 - \SampleData\PegaFFS_SnapDataInstances.jar
 4. **Optional:** Rulesets for IHS MarkIt and Equifax InterConnect have been removed from application ruleset stack in 8.3 release and rulesets for Refinitiv AVOX Legal Entity and eIdentification have been removed from application ruleset stack in 8.4 release. They have been compiled into components. If you used IHS MarkIt, Equifax InterConnect, eIdentification and/or Refinitiv AVOX Legal Entity in prior releases, please click on **Manage components** button in the application rule and check **Enabled** box next to the desired component.

Important: IHSMarKit and Equifax InterConnect components have been bundled into 8.3, 8.4, 8.5, 8.6, 8.7, and 8.8 releases only. Refinitiv AVOX Legal Entity and eIdentification components have been bundled into 8.4, 8.6, 8.7, and 8.8 releases only. To ensure that latest version components are available, customer will be downloading them from Pega Exchange/ Pega Marketplace in the future releases.

A new component to support Equifax's OneView offering is uploaded to Pega Marketplace. For more information, see OneView user guide.

Optional: Enabling sample operator accounts

Pega Foundation for Financial Services includes several sample operator accounts that are disabled for security purposes. Before you can use the sample application, enable a sample operator account.

1. Click **Configure > Org & Security > Authentication > Operator Access**.

2. In the Disabled operators section, click the check box next to the operator ID to enable, and then click **Enable selected**.
3. In the Enable operator dialog box, click **Submit** to confirm that you want to enable the selected operator ID.
4. Click **OK** to close the dialog box.

Optional: Components



Important: Download components for Financial Services (Refinitiv AVOX Legal Entity, Refinitiv World-Check One, Equifax InterConnect, IHS Markit, eldentification, and Dun & Bradstreet Direct API, Equifax OneView) if needed from Pega Marketplace.

Update and Hotfixes

- [Pega Foundation for Financial Services update guide](#)
- [Pega Foundation for Financial Services hotfixes](#)

Pega Foundation for Financial Services update guide

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

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`
 - `pegarules.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.

Completing the prerequisite tasks

Before you update to the latest release of Pega Foundation for Financial Services, complete the following tasks.

1. Install or update to Pega Platform '23 or the latest patch release.
For more information, see the [Pega Platform Installation Guide](#) or the [Pega Platform Update Guide](#) for your environment on the [Pega Platform page on Pega Community](#).
Patch releases are available through Pega's standard software delivery process. You can request software through Pega Software Distribution or by contacting GCS.
2. Update the online help to the latest version from the Online Help Files section on the Pega Platform Update page. After you download the latest `prhelp.war` file, deploy it in your installation before continuing. For deployment information, see the [Pega Platform Installation Guide for your application server](#).
3. Follow the update path instructions as mentioned below:
Updating from release 7.17 or 7.21: Updating from older versions of Pega Foundation for Financial Services is a multi-step process, therefore updating directly from release 7.17 to 8.23 is not recommended due to schema updates in releases 7.21 and 7.21.1. Follow these steps for the suggested best approach:
 - Upgrade from release 7.17 to 7.21
 - Upgrade from release 7.21 to 7.21.1
 - Upgrade from release 7.21.1 to 8.23

For more information, see the *Pega Foundation for Financial Services Update Guide* for a previous release on the [Pega Foundation for Financial Services product page](#) before proceeding further.

Updating from 7.21.1, 7.22, 7.31, 7.32, 7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8: You can update to release 8.23 directly from 7.21.1, 7.22, 7.31, 7.32, 7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, or 8.8. If you are updating to Pega Foundation for Financial Services '23 from release 7.21.1, 7.22, 7.31, 7.32, 7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, or 8.8 and plan to update the sample data after updating the application, drop the sample and snapshot tables before importing the application.

4. Review the database policies and application permissions that are used by your instance of Pega Platform to determine whether the application is permitted to update the database automatically, or if you must generate the database scripts that your organization will use make schema changes manually.

Updating the application

To update Pega Foundation for Financial Services from an earlier release, complete the following tasks.

Before you begin:

- Make sure that you back up your system and that you complete all the prerequisite tasks. For more information, see [Backing up your system](#) and [Completing the prerequisite tasks](#).
- Make sure that you become familiar with the list of removed or deprecated rules and properties. Additionally, if you are updating from release 7.4 or earlier, make sure that you become familiar with the list of removed classes.

For more information, see the *Pega Foundation for Financial Services Release Notes* that are available on the [Pega Foundation for Financial Services product page](#).

Preparing to update sample data

If you are updating to Pega Foundation for Financial Services '23 from release 7.21.1, 7.22, 7.31, 7.32, 7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, or 8.8 and plan to update the sample data after updating the application, drop the sample and snapshot tables before importing the application.



Note: If you do not plan to update the sample data and database schema, you can skip this task.

With Pega Foundation for Financial Services running on your application server, use the appropriate database tool to connect to the database, and then run the SQL script files that are appropriate for your database platform and schema.



Note: If you have a split-schema installation of Pega Platform, but use non-default schema names, update all references to PEGADATA in the applicable script before you run it.

The \ResourceKit\UpgradeScripts\PFFS\ directory on the distribution image contains several subdirectories that are specific to each supported database platform. The drop sample and snapshot table scripts are organized in the following structure:

- \DB2LUW\...
 - Drop_FSIF_Sample_Tables_DB2LUW_SingleSchema.sql
 - Drop_FSIF_Sample_Tables_DB2LUW_SplitSchema.sql
 - Drop_FSIF_Snap_Tables_DB2LUW_SingleSchema.sql
 - Drop_FSIF_Snap_Tables_DB2LUW_SplitSchema.sql
- \DB2zOS\...
 - Drop_FSIF_Sample_Tables_DB2_zOS_SingleSchema.sql
 - Drop_FSIF_Sample_Tables_DB2_zOS_SplitSchema.sql

- Drop_FSIF_Snap_Tables_DB2_zOS_SingleSchema.sql
- Drop_FSIF_Snap_Tables_DB2_zOS_SplitSchema.sql
- \MS-SQL\...
 - Drop_FSIF_Sample_Tables_MS-SQL_SingleSchema.sql
 - Drop_FSIF_Sample_Tables_MS-SQL_SplitSchema.sql
 - Drop_FSIF_Snap_Tables_MS-SQL_SingleSchema.sql
 - Drop_FSIF_Snap_Tables_MS-SQL_SplitSchema.sql
- \Oracle\...
 - Drop_FSIF_Sample_Tables_Oracle_SingleSchema.sql
 - Drop_FSIF_Sample_Tables_Oracle_SplitSchema.sql
 - Drop_FSIF_Snap_Tables_Oracle_SingleSchema.sql
 - Drop_FSIF_Snap_Tables_Oracle_SplitSchema.sql
- \PostgreSQL\...
 - Drop_FSIF_Sample_Tables_PostgreSQL_SingleSchema.sql
 - Drop_FSIF_Sample_Tables_PostgreSQL_SplitSchema.sql
 - Drop_FSIF_Snap_Tables_PostGreSql_SingleSchema.sql
 - Drop_FSIF_Snap_Tables_PostGreSql_SplitSchema.sql

Importing the application

Use the Import wizard to import the system data and rulesets for Pega Foundation for Financial Services to Pega Platform.

1. Log in to Pega Platform at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.

For example:

`administrator@pega.com`

2. In the header of Dev Studio, click **Configure > Application > Distribution > Import** and perform the following actions:

- a. Depending on your web browser, click Browse or Choose File, select the \Rules\PegaFFS_Int.jar file from your distribution image, and then click Next.
- b. Ensure that you import all components by leaving the Enable advanced mode to provide granular control over the import process check box cleared, and then click Next.

Result:

The wizard identifies differences between the database schema of your current system and the schema of Pega Foundation for Financial Services '23.

- c. If the file that you are importing requires changes to the database schema, select Automatic or Manual, and then click Next.
If you select Manual, see [Viewing and applying schema changes](#).
 - d. Follow the instructions on the screen, and when the import is completed, click Done.
3. Repeat step 2 for the following files:
 - \Rules\PegaFFS.jar
 - \Rules\PegaFFS_Sample.jar
 - \Rules\PegaFFS_Schema.jar
 - \Rules\PegaFFS_DataInstances.jar
 4. Apply the required hotfixes by using Hotfix Manager.
For more information, see [Pega Foundation for Financial Services Hotfixes](#).
 5. **Optional:** Install the Pega Product Designer for Financial Services.
For more information, see the *Pega Product Designer for Financial Services Installation Guide* that is available on the [Pega Foundation for Financial Services product page](#).
 6. **Optional:** For the best performance and user experience, pre-assemble the rules in the application by running the Static Assembler utility.

For more information, see [Preassembling rules in an application by using the Static Assembler utility](#).

7. In the **Explorer** panel, click Records > SysAdmin > RuleSet, and then verify that the *PegaFS* ruleset appears on the list.

Optional: Importing sample data

Pega Foundation for Financial Services includes some sample customer data that you can use to explore the application's functionality.

1. Log in to at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.

For example:

`administrator@pega.com`

2. In the header of Dev Studio, click Configure > Application > Distribution > Import and perform the following actions:
 - a. Depending on your web browser, click Browse or Choose File, select the `\SampleData\PegaFFS_SampleSchema.jar` from your distribution image, and then click Next.
 - b. Ensure that you import all components by leaving the **Enable advanced mode to provide granular control over the import process** check box cleared, and then click Next.

Result:

The wizard identifies differences between the database schema of your current system and the schema of Pega Foundation for Financial Services '23.

- c. If the file that you are importing requires changes to the database schema, select Automatic or Manual, and then click Next.

- If you select **Manual**, see [Viewing and applying schema changes](#).
- d. Follow the instructions on the screen, and when the import is completed, click **Done**.
 3. Repeat step 2 for the following files:
 - \SampleData\PegaFFS_SampleDataInstances.jar
 - \SampleData\PegaFFS_SnapSchema.jar
 - \SampleData\PegaFFS_SnapDataInstances.jar
 4. **Optional:** Rulesets for IHS MarkIt and Equifax InterConnect have been removed from application ruleset stack in 8.3 release and rulesets for Refinitiv AVOX Legal Entity and eIdentification have been removed from application ruleset stack in 8.4 release. They have been compiled into components. If you used IHS MarkIt, Equifax InterConnect, eIdentification and/or Refinitiv AVOX Legal Entity in prior releases, please click on **Manage components** button in the application rule and check **Enabled** box next to the desired component.

Important: IHSMarKit and Equifax InterConnect components have been bundled into 8.3, 8.4, 8.5, 8.6, 8.7, and 8.8 releases only. Refinitiv AVOX Legal Entity and eIdentification components have been bundled into 8.4, 8.6, 8.7, and 8.8 releases only. To ensure that latest version components are available, customer will be downloading them from Pega Exchange/ Pega Marketplace in the future releases.

A new component to support Equifax's OneView offering is uploaded to Pega Marketplace. For more information, see OneView user guide.

Optional: Enabling sample operator accounts

Pega Foundation for Financial Services includes several sample operator accounts that are disabled for security purposes. Before you can use the sample application, enable a sample operator account.

1. Click **Configure > Org & Security > Authentication > Operator Access**.

2. In the Disabled operators section, click the check box next to the operator ID to enable, and then click **Enable selected**.
3. In the Enable operator dialog box, click **Submit** to confirm that you want to enable the selected operator ID.
4. Click **OK** to close the dialog box.

Pega Foundation for Financial Services hotfixes

The following tables list the required hotfixes required by the Pega® Foundation for Financial Services application. To request a hotfix, go to your account on [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 Foundation for Financial Services installation or upgrade:

- Apply Pega Platform hotfixes immediately after the Pega Platform installation or upgrade.
- Apply Pega Foundation for Financial Services 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 Foundation for Financial Services 8.x**](#)
- [**Hotfixes for Pega Foundation for Financial Services 7.x**](#)

Hotfixes for Pega Foundation for Financial Services 8.x

The following tables list the required hotfixes required by the Pega® Foundation for Financial Services application. To request a hotfix, go to your account on [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.

- **Pega Foundation for Financial Services 8.8**
- **Pega Foundation for Financial Services 8.7**
- **Pega Foundation for Financial Services 8.6**
- **Pega Foundation for Financial Services 8.5**
- **Pega Foundation for Financial Services 8.4**
- **Pega Foundation for Financial Services 8.3**
- **Pega Foundation for Financial Services 8.2**
- **Pega Foundation for Financial Services 8.1**

Pega Foundation for Financial Services 8.8

Required Pega Foundation for Financial Services 8.8 hotfixes

Hfix ID	Pega Platform Version	Observed behavior
HFIX-85839	8.8	<p>As part of this Hfix, the following improvements are done to Target Operating model feature:</p> <ul style="list-style-type: none"> • Ability to automate the manual process of unit creation when WQs are created at Div level in Org chart. • Ability to automate the manual process of unit

Hfix ID	Pega Platform Version	Observed behavior
		value updation in WQ record when the divisions are imported from Org record (that is created from OOTB LP).
HFIX-85682	8.8	<p>This package includes</p> <ul style="list-style-type: none"> • Introduction of new properties to support ACH requirements in Model Workflow use case in SD • Resolved number format exception in validate rule 'FinTxnDefault' • Custom Index table to store accounting transactions • Updates to filter conditions on RDs fetching transaction summaries
Hfix-85751	8.8	<p>This package includes</p> <ul style="list-style-type: none"> • Integration layer properties to support ACH requirements in

Hfix ID	Pega Platform Version	Observed behavior
		<p>Model Workflow use case in SD</p> <ul style="list-style-type: none"> • Addition of new columns in tran and CC tran sample tables <div style="background-color: #e0f2ff; padding: 10px; margin-top: 10px;"> (i) Note: This is optional for customers as this package includes sample table updates </div>
Hfix-85750	8.8	<p>Sample data instances to validate ACH File functionality in Model Workflow use case in Smart Dispute application.</p> <div style="background-color: #e0f2ff; padding: 10px; margin-top: 10px;"> (i) Note: Hfix-85751 is pre-requisite before installing this Hfix. This is optional as this package contains sample data instances in transaction tables. </div>

Pega Foundation for Financial Services 8.7

Required Pega Foundation for Financial Services 8.7 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.7	Hfix-83975	This hotfix is delivered to support SD release 8.7.
8.7	Hfix-84020	This hotfix includes adding column to PegaFS-Int-FSF_Sample_Tran table to support SD release 8.7.

Note: This is optional for the customers as the updates are in sample table.

Pega Foundation for Financial Services 8.6

Required Pega Foundation for Financial Services 8.6 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.6.2	HFix-82802	Hotfix is delivered for exception handling to avoid infinite calls to the function 'AssessRelNetRecursiveOwnership' when the partyID is null.

Pega Platform Version	Hotfix Number	Observed Behavior
8.6.1	HFx-81590	<p>This hotfix is delivered to fix:</p> <ul style="list-style-type: none"> • System slowness due to data repletion in the Blob of Operating structure. • Fixing out the irregularities in indices creation whenever work group(s) is(are) added to a node (Div/ Unit).
8.6	HFx-84757	<p>This hotfix is delivered to backport the T.O.M features such as conversion of Org chart to Operating Structure, Importing of divisions to Operating Structure post conversion to PFFS r8.6.</p>
	HFx-82604	<p>This hotfix is delivered to resolve the issue that user is able to work on WB for which they do not have access.</p>
	HFx-82999	<p>This hotfix is delivered to introduce new property matchstatuslabel which will</p>

Pega Platform Version	Hotfix Number	Observed Behavior
		resolve issues around match status mapping.
	HFix-82605	This hotfix is delivered to fix StepNumber is not increasing for any new accounting.
	HFix-80732	<p>This hotfix contains the following changes.</p> <ul style="list-style-type: none"> • Risk classification for products in product matrix. • RPNewGeneration/maxDepthRelNet DSS. • LP_SP&LP_GE: Localization. • Regulatory details: Localization. • PegaFSIF access group changes.
	HFix-80751	<p>This hotfix contains the following changes.</p> <ul style="list-style-type: none"> • Localization of Satisfaction Logic display (SatisfactionLogicDescriptor property) in requirement set.

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> • Record for Monitor Doc Collection in case history localized.

Pega Foundation for Financial Services 8.5

Required Pega Foundation for Financial Services 8.5 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.5.5	HFx-83100	Hotfix is delivered to resolve issues around Date shift utility in PFFS.
	HFx-81781	This hotfix addresses the issue to enable download of uploaded document in Requirement cases
8.5.3	HFx-81208	This hotfix addresses the class inheritance for class Index-PegaAcct-WorkParty.
	HFx-81349	This hotfix addresses the Locking issue during Requirement Case(s) synchronisation.
	HFx-81208	Fix to execution of report definition (pyFollowedCases) for Smart Dispute (SD).

Pega Platform Version	Hotfix Number	Observed Behavior
		Please note that this is a required HFix for Smart Dispute application.
	HFix-81342	Hotfix is shipped to fix the irregularities in indices creation whenever work group(s) is(are) added to a node (Div/Unit) in Operating Structure.
	HFix-80729	This hotfix is shipped to correct the data repletion in blob of Operating Structure.
8.5.1	HFix-69148	This hotfix contains the schema and rule changes for EDR processing. This is required only for CLM/KYC applications.
	HFix-68460	<p>This hotfix contains the following changes.</p> <ul style="list-style-type: none"> • Upgrade orchestration changes. • Master profile for related party changes. • Indexes for operating structure nodes issue fix.

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> Additional data changes and field values.
8.5	HFx-82741	This Hotfix is delivered to fix the issue full name is more than 100 characters.

Pega Foundation for Financial Services 8.4

Required Pega Foundation for Financial Services 8.4 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.4.6	HFx-82414	<p>Following is the issue addressed as part of this hotfix:</p> <p>Column population job (triggered after optimising new properties) for PegaFS-Data-Party-MasterProfile class will pass whenever we optimize any property to the class.</p>
8.4.4	HFx-80612	Record for Monitor Doc Collection in case history localized.
	HFx-80511	Localization of Satisfaction Logic display (SatisfactionLogicDecoded property) in requirement

Pega Platform Version	Hotfix Number	Observed Behavior
		set. guide/86/step16-medical-dictionary-regulatory-activities-meddra-code-set.
8.4.3	HFix-81199	This hotfix addresses the class inheritance for class Index-PegaAcct-WorkParty.
	HFix-67642	Changes relating to the B20 security update.
	HFix-83972	This hotfix is shipped to remove XML elements that are validating the properties other than the ones listed in <i>FinTxnDefault</i> validate rule.
8.4.2	HFix-65660	Changes relating to the B20 security update.
8.4.1	HFix-81931	This hotfix addresses audit display issue with maintain case whenever page group properties are updated/modified.
8.4	HFix-61145	Class definition setting change to prevent history record of PegaFS-Data-ChangeTracker class from being written to pr_history table. It fixed issue where record id with 'HISTORY-' appended to it caused

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>insert failure into pr_history due to value exceeding column limit.Note: ChangeTracker class/table is used for auditing of value changes for properties.</p>

Pega Foundation for Financial Services 8.3

Required Pega Foundation for Financial Services 8.3 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.3.4	HFix-60643	Changes relating to the B20 security update.
8.3	HFix-67643	Localization of Satisfaction Logic display (SatisfactionLogicDecoded property) in requirement set.
	HFix-66838	In Pega Foundation for Financial Services, security (ASI) has been enhanced on several rules by requiring new privileges. The following out-of-the-box roles have been enhanced with the new privileges so the appropriate users can

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>access these enhanced rules.</p> <p>Roles: FSIF: Manager and PegaFSIF: User</p> <p>Privilege: AsyncProcessAccess- Privilege required to check if the asynchronous processing is completed or not. (Example: This is used in CLM application where async process is used in Capture and Enrich stage.)</p> <p>Privilege: DoSaveAccess- Privilege required to invoke action for saving a page (work page).</p>
	HFix-58197	<p>Avox is a third-party provider of enrichment data/report for organizations. Pega Foundation for Financial Services application uses Avox REST service (via calls) to retrieve organizational data.</p>

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>Changes to REST service connection rules for Authentication, Record Information for Organization and Subscription for Record Information.</p> <p>Change to formatting of Created Date for service call request to be compatible with international (operator) locales.</p>
	HFix-56626	<p>This hotfix is for CLM/KYC application and it includes:</p> <ul style="list-style-type: none"> • Data updates required to Relation code category/relation code reference data (tables): - FSF_RELCODES - FSF_PARTYRELATTYPE • Sorting retrieved relation codes alphabetically in descending order: - PARTYRELCODEXREFS UMMARYMAP (data transform) - GETPARTYIDRELCODES (report definition)

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> • Fixed Workgroups and Workbaskets for KYC/ CLM: - UPFS_GM_EMEA_GE_L KYC (workgroup) - UPFS_GM_EMEA_GE_R ELMGMT (workgroup) - BUSSPONSOR REVIEW (workbasket) - UPFS_GCB_FF (workbasket) • Updated class instance: - PEGAFS- DATA-PARTY- MASTERPROFILE

Pega Foundation for Financial Services 8.2

Required Pega Foundation for Financial Services 8.2 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.2.8	HFix-68067	Changes relating to the B20 security update.
8.2.1	HFix-51879	This hotfix includes the improvements added post Pega Foundation for Financial Services 8.2 release. It also includes the optional sample data capability.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-54284	AVOX provider fix (when rules) and schema changes to FSF_PartyPartyXref table with new properties and report definition update.
8.2	HFx-56094	Fix for Risk Factor Change Tracking. Recording of property value changes in PFFS Change Tracking table.
	HFx-56076	This hotfix provides fix for Risk Factor value change tracking as recorded in FSF_Changetracker table.

Pega Foundation for Financial Services 8.1

Required Pega Platform hotfixes for Foundation for Financial Services 8.1

Pega Platform Version	Hotfix Number	Observed Behavior
8.1.9	HFx-67873	Changes relating to the B20 security update.
8.1	HFx-46785	Fixed data type reversal of 2 columns pyslaaction and pyworklistdate1 , in table paf_work

Note: This hotfix must be imported before upgrading to Pega

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>Foundation for Financial Services 8.1.</p>

Required Pega Foundation for Financial Services 8.1 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.1	HFix-58178	Fixed select file button for attachment functionality (pyUploadFile rule).
	HFix-58034	Fixed navigation summary to display alternative stage step as enabled in the case.
	HFix-57500	The getStateList activity that retrieves a list of US states has been fixed to remove recursive call to D_StateList. Note: getStateList/D_StateList is restrictive to only US states and as a result D_StateNameSummary was introduced. It provides international states/province names and codes based on the country code (Ex: US) inputted. Its recommended that

Pega Platform Version	Hotfix Number	Observed Behavior
		customers use D_StateNameSummary.
	HFix-53143	Operational Structure: Fixed FSOrgChartShow activity to map Organization Unit, Division and Organization rules to clipboard.
	HFix-52385	Requirement case routing fix: Modified ProcessRequirement activity to set the operator assignment to current operator if it is automated case otherwise route to worklist or workbasket.
	HFix-52070	Fixed adding new funds while running onboarding case with organization. Changes done in the declare expression to execute D_SubIndustryDataOpen datapage only if the parameter value is not blank.
	HFix-51763	SatisfactionLogicDecoded is set with a translation of logical expression for labeling purposes in the UX for requirement case.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-51762	Requirement Validation: Modified the ValidateEachReqDoc activity to validate status property before setting error message on the UI.
	HFx-51575	Fix to save the modified documents via Requirements Portal.
	HFx-51477	FieldValues (reference data) for UX Localization such as country names and code descriptions.
	HFx-50596	Section rule pyDisplayStages withdrawn to enable case stages work with UI kit 12.01.
	HFx-49595	Added properties and updated sample data to fix missing NZ account in 'account selection' page.
	HFx-48980	Support for "Failed to Install Pre-requisite PFFS 8.1HFx-48041".
	HFx-48041	This hotfix provides a fix to DownloadContentStream activity to call Webstorage/ Repository.

Hotfixes for Pega Foundation for Financial Services 7.x

- [Pega Foundation for Financial Services 7.4](#)
- [Pega Foundation for Financial Services 7.32](#)
- [Pega Foundation for Financial Services 7.31](#)
- [Pega Foundation for Financial Services 7.22](#)
- [Pega Foundation for Financial Services 7.21.1](#)
- [Pega Foundation for Financial Services 7.21](#)
- [Pega Foundation for Financial Services 7.17](#)
- [Pega Foundation for Financial Services 7.16](#)
- [Pega Foundation for Financial Services 7.15](#)
- [Pega Foundation for Financial Services 7.14](#)

Pega Foundation for Financial Services 7.4

Required Pega Platform hotfixes for Foundation for Financial Services 7.4

Pega Platform Version	Hotfix Number	Observed Behavior
7.4	HFx-67782	Changes relating to the B20 security update.
	HFx-54875	<p>The following changes are addressed in the following hotfix:</p> <ul style="list-style-type: none"> • Created Merchant Category Codes look up table (reference and snap) and data.

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> Added Merchant Category and Code columns to Authorization table. Updated sample data for Authorization and CCTran (credit card transactions) tables. Primary application user: Smart Dispute (SD).
	HFix-46786	<p>Fixed data type reversal of 2 columns pyslaaction and pyworklistdate1, in table paf_work.</p> <p>Important: This hotfix must be imported before upgrading to Pega Foundation for Financial Services 7.4.</p>
7.3.1	HFix-46785	<p>Fixed data type reversal of 2 columns pyslaaction and pyworklistdate1, in table paf_work.</p> <p>Important: This hotfix must be imported before upgrading (Pega Platform 7.3.1 + PFFS 7.32 to Pega Platform 7.4 + PFFS 7.4). This hotfix is not required for fresh installation.</p>

Pega Platform Version	Hotfix Number	Observed Behavior
7.3	HFx-46486	<p>Fixed data type reversal of 2 columns pyslaaction and pyworklistdate1, in table paf_work.Important: This hotfix must be imported before upgrading (Pega Platform 7.3 + PFFS 7.31 to Pega Platform 7.4 + PFFS 7.4). This hotfix is not required for fresh installation.</p>

Required Pega Foundation for Financial Services 7.4 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
8.1	HFx-47925	This hotfix is for Pega Product Designer Financial Services. No open ruleset warning/issue - rules have been updated to point updated Pega Platform API to get latest rule set version at run time.
	HFx-47919	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> • Search with phone number is not working as expected.

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> Documents upload is not working as expected.
7.4	HFix-58761 HFix-58232	<p>APPAMOUNTWITHCURRENCY input rule was updated to allow input of amount 0.00. Before the change, validation didn't allow amount <= 0. Updated condition is as follows: if amount < 0 then output error message otherwise pass.</p> <p>The getStateList activity that retrieves a list of US states has been fixed to remove recursive call to D_StateList.</p> <div style="background-color: #e0f2ff; padding: 10px; border-left: 2px solid #336699; margin-left: 20px;"> Note: getStateList/ D_StateList is restrictive to only US states and as a result D_StateNameSum mary was introduced. It provides international </div>

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>states/province names and codes based on the country code (Ex: US) inputted. Its recommended that customers use D_StateNameSummary.</p> <p>(i)</p>
	HFix-56982	<p>Accounting: Implemented check to ensure that memo inserted into pc_link_attachment table is not over 64 chars long.</p> <p>If longer then truncated to 64 characters.</p> <p>(i)</p> <p>Note: Full description (memo) is still inserted into pc_data_workattach table. Primary user of this functionality is SD.</p>

Pega Platform Version	Hotfix Number	Observed Behavior
	HFH-56075	Pega Accounting Overall SLA escalation action fix to increase item's work urgency (FinUrgencyUpdateAssignments activity).
	HFH-53519	Currency formatting was being hardcoded as USD and therefore all transaction values were formatted as USD (rounding). EX: If customer had JPY then it was still formatting as USD. Base currency is now (dynamically) set based on passed parameter that can represent currency of the transaction. If parameter is empty then currency formatting defaults to USD.
	HFH-50567	Updated case of property to fix case sensitivity of a property causing incorrect behaviour.
	HFH-49590	Fixed country data modification for CRS and IGA and added addition properties for CRS and IGA.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-49144	The AccountNumber unmasked while printing case details.
	HFx-48954	The Test Case screen is not working as expected.
	HFx-48619	Empty Assignment key error in WSS portal.
	HFx-48541	SearchDocuments activity fixed to explicitly create and delete temp page.
	HFx-48371	DSER : Table Indexes and property optimisation.
	HFx-47174	Provide LocalPartySubtypes Schema Changes.
	HFx-46933	Unable to view Full Notes and Correspondence in History.
	HFx-46794	Request to include customer account.
	HFx-46789	Request to package new requirement changes.
	HFx-46344	Unable to run Date Shift.
	HFx-45487	Feedback For WEB Storage in PFFS 07.4.
	HFx-45376	Currency changes for Venezuela,Mauritania, Sao Tome and Principe.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-45243	Support for "Too much audit info logged for Accounting Steps".
	HFx-44791	Properties to be created and exposed in master profile.
	HFx-44352	Fixed conditions to allow Web storage/CMIS file upload. Created extension rule SETCMISDOCMETADATAEXT for custom functions by consuming apps. Fixed creation of new requirement set to When Satisfaction rule generation. Important: This hotfix must be imported after HFx-44191 .
	HFx-44191	This hotfix provides Hotfix Manager Support for Pega Foundation for Financial Services 7.4 application.
	HFx-44118	This hotfix provides fixes of defects for CLM.

Pega Foundation for Financial Services 7.32

Required Pega Foundation for Financial Services 7.32 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.4	HFH-43732	<p>Changes are required to address the following issues:</p> <ul style="list-style-type: none"> Customer search is not returning records. When updating document, message was incorrectly showing a successfully added instead of successfully updated. Newly added accounts not displayed in composite view. Complete stage button incorrectly shown when adding a financial goal.
	HFH-40384	Country names not consistent among the data pages.
	HFH-40293	Fixes related to eScreening and additional sample data.
	HFH-41827	Fix fetching of data from external search providers in French.
	HFH-39801 (DL-78224)	When a user tries to add an existing related party, data

Pega Platform Version	Hotfix Number	Observed Behavior
		is not fetched and the fields are blank and editable.
7.3.1	HFHix-44353	Fixed conditions to allow file upload to CMIS/web storage. Created extension rule SETCMISDOCMETADATAEXT for custom functions by consuming apps.
	HFHix-41827	Fix fetching of data from external search providers in French.
	HFHix-39801 (DL-78224)	When a user tries to add an existing related party, data is not fetched and the fields are blank and editable.
7.3	HFHix-45489	This hotfix provides Feedback support for Web Storage.
	HFHix-43113	OOTB agents running RD's on unexposed columns.
	HFHix-42087	Update sample data to include Croatia and Malta in EU Jurisdiction.
	HFHix-41778	Set dynamic system setting to false to allow 3rd party web services connections.

Pega Platform Version	Hotfix Number	Observed Behavior
7.2.2	HFx-41778	Set dynamic system setting to false to allow 3rd party web services connections.

Pega Foundation for Financial Services 7.31

Required Pega Foundation for Financial Services 7.31 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.3.1	HFx-35698	A user interface (UI) screen display issue effects Customer Search functionality.
	HFx-36556	Pega Foundation for Financial Services 7.31 requires changes and additions to the Requirements (Document requirements) case type. Additional changes to sample data are included to maintain enhanced country/ state information, product type information in product matrix table, and account information in deposit account table. All changes support the latest Pega Client Lifecycle Management for Financial Services and Pega Know

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>Your Customer for Financial Services functionality. Note: This hotfix is applicable for Pega Foundation for Financial Services 7.31 installations or upgrades when you install or upgrade Pega Client Lifecycle Management for Financial Services 7.31 and Pega Know Your Customer for Financial Services 7.31.</p>
	HFix-37339	<p>This hotfix is required to address the changes in Requirements Case Type, Data Traceability and Screening error handling.</p> <div style="background-color: #e0f2ff; padding: 10px; margin-top: 10px;"> <p>Note: This hotfix is not required if HFix-36556 is already imported.</p> </div>
	HFix-38959	<p>As a part of Know Your Customer Regulatory Compliance 17.3 release, 18 new KYC types have been added which require new accounts to be included</p>

Pega Platform Version	Hotfix Number	Observed Behavior
		from Pega Foundation for Financial Services.
	HFix-39363 (DL-77415)	Customers are unable to proceed further after uploading the documents.
	HFix-39809	Know Your Customer Regulatory Compliance 17.3 release requires additional sample data for 18 new KYC types. And also modified data transform to map country codes.
7.3	HFix-45488	This hotfix provides Feedback support for Web Storage.
	HFix-41716	The "PYPARTYFULLNAME" is setting as null in the "pc_index_workparty" table.
	HFix-41657	Fix fetching of data from external search providers in French.
	HFix-38929	Reversion to generically applicable legal hierarchy calculation.
	HFix-36556	Pega Foundation for Financial Services 7.31 requires changes and additions to the

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>Requirements (Document requirements) case type.</p> <p>Additional changes to sample data are included to maintain enhanced country/state information, product type information in product matrix table, and account information in deposit account table. All changes support the latest Pega Client Lifecycle Management for Financial Services and Pega Know Your Customer for Financial Services functionality.</p> <p>Note: This hotfix is applicable for Pega Foundation for Financial Services 7.31 installations or upgrades when you install or upgrade Pega Client Lifecycle Management for Financial Services 7.31 and Pega Know Your Customer for Financial Services 7.31.</p>

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>Customer for Financial Services 7.31.</p>
	HFix-35698	<p>A user interface (UI) screen display issue effects Customer Search functionality.</p>

Pega Foundation for Financial Services 7.22

Required Pega Foundation for Financial Services 7.22 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.3	HFix-35605	WriteOff links are not available for the Accounting Flow.
7.2.2	HFix-80631	Changes relating to the B20 security update.
	HFix-46146	This hotfix provides solution for 'when' condition to call Sava to Web Storage activity.
	HFix-45490	This hotfix provides Feedback support for Web Storage.
	HFix-45373	Currency changes for Venezuela, Mauritania, Sao Tome and Principe.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFH-44917	Issue while downloading attached files.
	HFH-44252	Property Field Value validation on MultipleClearingSeqNumber .
	HFH-44012	This hotfix resolves the CMIS connectivity error.
	HFH-39905	The "RetrievalReferenceNumber" field needs to be added in CCTran table.
	HFH-37618	New database column needed in FSIF_SAMPLE_CCTRAN table.
	HFH-37317	The FNS Counter not being set.
	HFH-36968	Changes are required to allow Pega Foundation for Financial Services to integrate with Content Management Interoperability Services (CMIS) and Documentum server.
	HFH-36414	This hotfix is prerequisite for FSIF 7.22 HotFix manager support.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-35605	The WriteOff Links is unavailable for the Accounting Flow.
	HFx-35434	The Pega Smart Dispute for Financial Services application needs the latest sample data for Visa Claims Resolution (VCR) Patch3.
	HFx-31046	A utility is required to move requirements objects/work cases into mainstream processing. This was needed due to functionality conflicts between Pega Foundation for Financial Services 6.1 Sp2 and Pega Foundation for Financial Services 7.11.
	HFx-32737	All mandatory fields do not display under the Pega Know Your Customer Completion section.
	HFx-33653	The Customer Relationship Management for Financial Services 7.22.1 release requires this hotfix in order to take advantage of the latest Pega Foundation for Financial Services layer enhancements which

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>includes the following changes:</p> <ul style="list-style-type: none"> • Updates to Data-Admin-Product-Framework class instances that support the Hotfix Manager feature. • New functionality to support consuming apps: Data Traceability and WC integration. • Support for KYC-722 Compliance using 16.4-Multi Application testing. • For DB2 environments, updates to address errors observed upgrading earlier versions of Pega Foundation for Financial Services (7.15, 7.16, 7.17, 7.21) to Pega Foundation for Financial Services 7.22. • A Sample Data update is required to support Customer Relationship

Pega Platform Version	Hotfix Number	Observed Behavior
		<p>Manager for Financial Services 7.22.1.</p> <div style="background-color: #e0f2ff; padding: 10px; border-radius: 5px;"> <p>Note: View the ReadMe for details about executing the Sql script prerequisite prior to importing the JAR files in the hotfix package.</p> </div>

Pega Foundation for Financial Services 7.21.1

Required Pega Foundation for Financial Services 7.21.1 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.2.2	HFix-80411	<p>Following is the issue addressed as part of this hotfix:</p> <p>Setting values for stepPage and workPage in FinWriteEntry activity(PegaAcct-Txn-).</p>
	HFix-37162	Changes are required to allow Pega Foundation for Financial Services to integrate with Content

Pega Platform Version	Hotfix Number	Observed Behavior
		Management Interoperability Services (CMIS) and Documentum server.
7.2.1	HFix-67875	Changes relating to the B20 security update.
	HFix-59908	For Smart Dispute (SD): MerchantCountry column and data was added to tables FSF_REF_MERCHANTCATEG ORY and FSF_SAMPLE_AUTHORIZATI ON.
	HFix-49143	AccountNumber unmasked while printing case details 7.22.
	HFix-45491	Feedback for Web Storage.
	HFix-45375	Currency changes for Venezuela,Mauritania, Sao Tome and Principe.
	HFix-44242	Property Field Value validation on MultipleClearingSeqNumber II.
	HFix-44856	Update the transaction type column in cctran.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-43736	Transaction Description of length more than 36 causing error.
	HFx-42281	Currency Code is not appearing properly.
	HFx-40228	The "RetrievalReferenceNumber" field needs to be added in CCTran table.
	HFx-38586	Scoring Definition-DuplicateScore rules changes.
	HFx-37400	This hotfix is prerequisite hotfix for FSIF 7.21 to provide ML1 HotFix manager support.
	HFx-37162	Unable to use CMIS integration with Documentum.
	HFx-33432	Updates to Data-Admin-Product-Framework class instances that support the Hotfix Manager feature.
	HFx-30270	Changes are required to address the following issues: <ul style="list-style-type: none"> • Date Shift utility is not working as expected.

Pega Platform Version	Hotfix Number	Observed Behavior
		<ul style="list-style-type: none"> Addressed document collection requirements. Removed the Test id/password provided with Equifax and Markit Connectors.
	HFix-29871	Cascading error message in Manage communication preferences fixed with this base Pega Platform hotfix.

Pega Foundation for Financial Services 7.21

Required Pega Foundation for Financial Services 7.21 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.2.1	HFix-56076	Pega Accounting OverallSLA escalation action fix to increase item's work urgency (FinUrgencyUpdateAssignments activity).
	HFix-37617	New DB Column needed in FSIF_SAMPLE_CCTRAN table.
	HFix-36756	Fin Verification Flow Action Pre-processing Error.
	HFix-36694	The FNS Counter is not being set.

Pega Platform Version	Hotfix Number	Observed Behavior
7.1.9	HFH-44631	The performance issue with OOB activity "FinWriteEntry".

Pega Foundation for Financial Services 7.17

Required Pega Foundation for Financial Services 7.17 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.2.2	HFH-36645	Changes are required to allow Pega Foundation for Financial Services to integrate with Content Management Interoperability Services (CMIS) and Documentum server.
7.2.1	HFH-39886	The assignment info is not updated when case is in Screenflow.
	HFH-26625	Changes for Pega Onboarding for Financial Services, Data Page, Sample Data.
	HFH-27071	Changes for Pega Onboarding for Financial Services, Data Page, Sample Data.
	HFH-27491	Prospect search is not working.

Pega Platform Version	Hotfix Number	Observed Behavior
7.2	HFHix-27329	Cannot Attach Requirement Document for Multiple Parties.
	HFHix-28533	Addition of the WalletProviderID to the data model.
	HFHix-27970	Error is throwing upon clicking service customer link.
	HFHix-28010	Markit flow functionality is not working properly.
	HFHix-28016	FNSCounter property value alignment.
	HFHix-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO) Bank Identification Numbers (BINs) beginning with a "2".
	HFHix-33432	Updates to Data-Admin-Product-Framework class instances that support the Hotfix Manager feature.
7.2	HFHix-39457	Remove RuleSet restriction for Class PegaFS-Work-CustomerSearch.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFx-37866	Case overview screen to show same casetype in multiple stages.
	HFx-26625	Changes for Pega Onboarding for Financial Services, Data Page, Sample Data.
	HFx-27071	Changes for Pega Onboarding for Financial Services, Data Page, Sample Data.
	HFx-27491	Prospect search is not working.
	HFx-27329	Cannot Attach Requirement Document for Multiple Parties.
	HFx-28533	Addition of the WalletProviderID to the data model.
	HFx-28016	FNSCounter property value alignment.
	HFx-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO) Bank Identification Numbers (BINs) beginning with a "2".

Pega Platform Version	Hotfix Number	Observed Behavior
7.1.9	HFx-36644	This hotfix is the prerequisite for FSIF 7.17 HotFix Manager Support.
	HFx-28533	Addition of the WalletProviderID to the data model.
	HFx-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO) Bank Identification Numbers (BINs) beginning with a "2".

Pega Foundation for Financial Services 7.16

Required Pega Foundation for Financial Services 7.16 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.21	HFx-37616	New DB Column needed in FSIF_SAMPLE_CCTRAN table.
7.2	HFx-25620	System is getting hanged upon clicking Manage Documents.
	HFx-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO)

Pega Platform Version	Hotfix Number	Observed Behavior
		Bank Identification Numbers (BINs) beginning with a "2".
7.1.9	HFx-33432	Updates to Data-Admin- Product-Framework class instances that support the Hotfix Manager feature.
	HFx-24134	Required additional fields in Acct and Tran table for ACH.
	HFx-24768	Issue in accounting rules and FSPortal.
7.1.7	HFx-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO) Bank Identification Numbers (BINs) beginning with a "2".
	HFx-28533	Addition of the WalletProviderID to the data model.
	HFx-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO) Bank Identification

Pega Platform Version	Hotfix Number	Observed Behavior
		Numbers (BINs) beginning with a "2".

Pega Foundation for Financial Services 7.15

Required Pega Foundation for Financial Services 7.15 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.1.7	HFix-28533	Addition of the WalletProviderID to the data model.
	HFix-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO) Bank Identification Numbers (BINs) beginning with a "2".
7.1	HFix-37612	New database column needed in FSIF_SAMPLE_CCTRAN table.

Pega Foundation for Financial Services 7.14

Required Pega Foundation for Financial Services 7.14 hotfixes

Pega Platform Version	Hotfix Number	Observed Behavior
7.1.6	HFix-28533	Addition of the WalletProviderID to the data model.

Pega Platform Version	Hotfix Number	Observed Behavior
	HFix-28736	Enhancement of sample data to the Smart Disputes support of MasterCard International Organization for Standardization (ISO) Bank Identification Numbers (BINs) beginning with a "2".
7.1	HFix-37609	New database column needed in FSIF_SAMPLE_CCTRAN table.

Product overview

- Pega Foundation for Financial Services features
- Roles, portals, and dashboards
- Primary data entities
- Further reading

Pega Foundation for Financial Services features

Pega Foundation for Financial Services provides a rich set of features designed to enhance the customer experience, improve user productivity and increase customer satisfaction.

This section describes key capabilities and features of the application that you can use as-is or extend to meet your business needs.

- Search and composite view
- Determining entity risk
- Information providers
- Onboard contracting party requirements
- Financial accounting
- Operational and product structures
- Financial services data
- Customer master profile

- Event driven architecture
- Async processing

Search and composite view

Pega Foundation for Financial Services can help your customers achieve their desired goal quickly and with minimal steps, ensuring that they complete the interaction positively and with great satisfaction.

Because searching for an individual or organization is a common activity, a set of interfaces is built within Pega Foundation for Financial Services that enables you to:

- Update information, such as addresses, communication preferences, customer details, financial details, travel notifications, automated payment details, and beneficiaries.
- Manage, create, or associate existing related parties and account relationships.
- Define a variety of customer's financial goals and track progress, showing summary results on the customer composite view.
- See the customer's risk profile and edit where appropriate.
- View compliance assessments and enriched data from external business services.
- See composite views that are tailored to support organizations and individuals.

Determining entity risk

The Risk Assessment is a declarative network that tracks a wide range of data elements that impact the customer risk. One or more elements are linked to a single risk factor such as Product, Country, or Related Parties. These lower level risk factors roll up to a single overall risk for the entity.

The application includes customer risk scores for both Individuals and Organizations and are leveraged by the client life-cycle management applications in determining document requirements.

Information providers

Enriching customer and company data from third-party providers enable financial services organizations many benefits including more efficiently managing credit decisions, verifying identity, increasing fraud awareness, providing more targeted and personalized cross-sell and upsell strategies, and optimizing collections treatments.

Pega's ecosystem of pre-built connections to providers are available on Pega Exchange and can be incorporated into Pega Foundation for Financial Services using a centralized set of objects.

The following are the pre-built connectors currently available from Pega Marketplace and have pre-built configurations with the Foundation for Financial Services centralized Business Service.

Service Provided	Provider Name
Screening of Organizations	Refinitiv World-Check One
Enrichment of Organizations	Dun & Bradstreet Direct API
Enrichment of Organizations	IHS Markit
Enrichment of Organizations	Refinitiv AVOX Legal Entity
Enrichment of Organizations	SWIFT KYC
Credit Profile of Individuals	Equifax OneView
Authenticate/Verify	eldentification

For more information on configuring these integrations, see the Pega Foundation for Financial Services [product page](#).

Onboard contracting party requirements

Requirements are used to help organizations manage multi-stage, complex document and requirement collection processes.

- Enables you to upload any required supporting documents as determined by the financial services application logic.
- Provides an interface for you to review, manage, and approve user documents.

For more information on configuring the requirement functionality as a user or developer, see the Pega Foundation for Financial Services [product page](#).

Financial accounting

Financial institutions need to track the transactions customers are performing, so Pega has built robust sub-ledger accounting that includes suspense and adjustment accounting, write-off management, multi- level verification, correspondence generation. The application tracks any changes to these accounting functions.

For more information on configuring the accounting functionality, see the Pega Foundation for Financial Services [product page](#).

Operational and product structures

You can create your organization's multi-level operational and product structures and easily generate the appropriate workgroups and workbaskets at the desired levels.

- Sales or onboarding are more efficient by only showing the appropriate list of products based on inclusion and exclusion logic that is created during the product design.
- If upgrading from a prior release, previously created workbaskets and workgroups can be used by the enhanced operational structure.
- Sample operational structures called PegaBank and Uplus Financial Services are provided along with a sample Uplus Financial Services product structure.

Operational structure

View operating structure for **UPlus Financial Services**

Organizational chart **Taxonomy**

Entity name	Level	Calendar	Currency	Departments	Financial business segment	Actions
UPlus Financial Services	L+ Add Financial Institution	UPFSDefault USD	0			
Asset Management	L+ Add Business Line	UPFSDefault USD	8	Corporate & Investment Banking		
Global Business Banking	L+ Add Business Line	UPFSDefault USD	0	SMB		
Global Consumer Banking	L+ Add Business Line	UPFSDefault USD	7	Retail		
Austria	L+ Add Country	UPFSDefault USD	0	Retail		
Vienna	L+ Add City	UPFSDefault USD	0	Retail		
Vienna Central Branch	Branch	UPFSDefault USD	0	Retail		
Australia	L+ Add Country	UPFSDefault USD	0	Retail		
Sydney	L+ Add City	UPFSDefault USD	0	Retail		
Sydney Central Branch	Branch	UPFSDefault USD	0	Retail		

Financial services data

Pega provides a comprehensive suite of adapters that enable Pega applications to fit seamlessly into your technology ecosystem. Pega supports transport or protocol-level adaptors with form-based integration options, so that no custom code is required to connect to an external system. Pega Live Data simplifies using data in business processes by delivering the right data, in the right place, at the right time. Data requests are managed behind the scenes so that data flows to the right process steps. It is easy to change and adapt for new data sources and new applications, making it available across all your Pega applications.

The customer journeys within Pega's Financial Services applications utilize an extensive set of financial and reference information, much of which will reside in your own systems of record. You decide which data should reside in Pega and which to reference from your existing sources.

More than 80 data classes have been defined for commonly referenced financial objects like customers, accounts, transactions, classifications, relationships, and so on.

The data model forms the enterprise backbone for all Pega financial services applications and supports most business functions in financial organizations.

Customer master profile

Customer master profile represents the local copy of an actual customer. The local copy of driving customer data is intended to keep in the system. As the customer related data resides in the financial institution systems of record (likely to be external to the application), the systems must rely on the local copy of driving data in order to function without depending on the external systems of record.

Driving data directly drives the Financial services customer journeys related to customer risk profiling, compliance elements (KYC profile information), jurisdictional elements, product related, relevant related parties, and so on.

Event driven architecture

Event driven architecture is a functional framework embedded within PFFS layer that facilitates implementation of event-driven processes. This feature manages the reception of events from different sources and route them to the appropriate applications and processes in charge of their resolution. Event driven architecture is designed as an infrastructural module. It does not implement any specific event or process. It provides a framework in which other modules can define their own events and processes and manage them in an efficient and consistent manner.

Async processing

Heavy processes that run during a user's session can create delay between the last action that the user took (for example, submitting an assignment) and the return of the control to the user, either by showing the confirm harness or by opening the next active assignment. For processes where the execution time takes a few seconds, and the current operator is likely to take the next action, this waiting period is acceptable. It is not acceptable, though, in cases where the execution time goes beyond ten or more seconds, and a different team likely performs the next task.

For processes where the waiting period is excessive, the system uses asynchronous processing. This gives control back to the users as soon as they take action on the case to keep working on other cases while the system processes the slower case. The system works asynchronously until human intervention is required or the case is resolved.

Roles, portals, and dashboards

Pega Foundation for Financial Services supports key roles associated with your day-to-day use of the application. The table below describes the key access roles provided with Pega Foundation for Financial Services

Role name	Description
Case Worker	Primary focus of case workers is searching for customers and helping with documentary requirements. Therefore, a portal showing assignments and tasks in order of urgency is provided for the case worker.
Case Manager	The case manager portal enables them to monitor the prioritized list of open cases, team members, and work. They can customize the portal to highlight the most important sections to them.
Business Analyst	Business analysts manage the documents required to sell products to customers and organizations. The requirements portal enables managing documents along with the conditions when they are required. The document requirements for onboarding and selling products can be complex, so there are multiple layers in which the business analyst

Role name	Description
	<p>can define applicability and satisfaction rules. These layers include:</p> <p>Requirement sets: can define the conditions in which a requirement set is applicable along with which requirements will need to be satisfied.</p> <p>Requirements: can define a requirement along with the documents that will need to be satisfied. When running a case, the requirement is created and linked to a stage in the processing flow of the parent case.</p> <p>Documents: can define new documents and associate them with a product or do the same for existing documents.</p> <p>A portal is included that helps manage these elements so that business analysts can manage the various jurisdictions rules.</p>

Further reading

For more information about Pega Foundation for Financial Services, see [product page](#).

Primary data entities

This section describes the data types that are used by Pega Foundation for Financial Services, stored in external systems or, where appropriate, internally to the Pega system.

Name	Description	System of record (SOR)
Accounts	Financial account information associated to an organization or individual that includes credit, debit, deposit, loan, and investment accounts.	External
Account transactions	Transactions occurring within credit, debit, loan, investment, and deposit accounts.	External
Contact	A person that is a prospect or customer. For business selling, they are usually employees of the organization to whom you are selling. For Individual selling, they are the person interested in making a personal purchase.	Pega or External
Customer	An individual or organization that has purchased product or services and holds accounts with the financial institution.	Pega or External
Individuals	An individual person who can be a customer, contact, or prospect to the financial institution.	Pega or External

Name	Description	System of record (SOR)
Households	A group of individuals related and aggregated together so that the optimal advice, offers, and actions are provided when household members are interacting with the Financial Institution.	External
Organizations	A business, holding company, or corporation that can be a customer, contact, or prospect to the financial institution. It can consist of one or more organizations that can hold many accounts. Some examples include: corporations, partnerships, associations, funds, trusts, family offices, and endowments.	Pega or External
Parties	A legal organization, individual, or fund that can be a customer, contact, prospect to the financial institution.	Pega or External
Products	Products being sold and serviced to prospects and customers as separate	External

Name	Description	System of record (SOR)
	<p>products or as part of bundles.</p> <p>Includes the product's pricing, eligibility, and benefits.</p>	
Prospect	An individual or organization that does not yet own products of the financial institution and thus is not yet a customer.	Pega or External

Further reading

For more information about Pega Foundation for Financial Services, see [product page](#).

Implement

- Pega Foundation for Financial Services implementation guide

Pega Foundation for Financial Services implementation guide

This guide helps you to have an easier and more focused experience to learn and understand how to implement the Financial Services foundation.

- Preparing for implementation
- Prerequisites
- Creating new application
- Defining the data model
- Access groups
- Connecting system of record
- Managing reference data
- Application configurations
- Integrations
- Defining the security model and organization structure
- Customizing the user experience
- Features implementation
- Implementing case types

Preparing for implementation

Built for financial services organizations, Pega Foundation for Financial Services provides common, reusable components on which applications, including other financial service applications, can be built.

Financial Services organizations can use Pega Foundation for Financial Services to:

- Address new market opportunities in a fraction of time required by more traditional development environments and accelerate the development of personalized solutions targeting your organization's biggest challenges across your key domains of marketing, sales and onboarding, customer service, and operations.
- Develop applications that dramatically extend or replace the capabilities of existing systems.
- Revitalize legacy applications.
- Create and maintain a layer of shared assets across your Pega application portfolio.

Pega Foundation for Financial Services forms the enterprise foundation for all Pega solutions.

*Application architecture*

Pega Foundation for Financial Services comprises the application layers described below. The individual application modules provide flexibility and speed of deployment by allowing you to install and use only the modules that you need, based on your envisioned use.

The following application bundles are included on your media:

- PFFS – The core foundation application that contains the financial services data model, requirements, accounting, and other features used within Pega's strategic applications. Installing this application is a pre-requisite for all the other optional application modules.
- **Optional** Pega Foundation for Financial Services sample application.

- **Optional** Schema and Sample data for customers, accounts and transactions.
- **Optional** Product Designer for Financial Services application for designing and managing products. Includes sample products.

Prerequisites

Before you create the baseline application, be sure that you install and test Pega Foundation for Financial Services. See the Pega Foundation for Financial Services Installation Guide on the Pega Foundation for Financial Services [product page](#).

Creating new application

- [Creating the baseline application](#)
- [Loading sample data](#)

Creating the baseline application

Use the Application Wizard to create a new application built on Pega Foundation for Financial Services. The wizard creates the application structure for you.

In some cases, an organization that consists of multiple companies or entities requires multiple applications to vary the features based on the business needs of each company or entity. If your application features are the same throughout your organization, create an implementation application only. If your application features vary for different parts of your organization, first create a framework enterprise application that contains the features that are common to all parts of the organization. Then, create an implementation application for each variation that your organization requires. When using a framework application, build the implementation applications on top of the framework application so that they inherit common rules from the framework application.

Creating the implementation application only

Pegasystems recommends that you extend from the Pega Foundation for Financial Services (PFFS) application and include the optional applications (listed in the Application Overview section earlier) in your application rule based on your envisioned solution.

If you are creating a new application leveraging one of the Pega Financial Services Applications (e.g. Customer Service, CLM, KYC, Smart Dispute and Sales Automation), you can skip this step and create your baseline application by following the steps included in the Implementation Guide of the said Application.

Running the new application wizard

To create an application on Pega Foundation for Financial Services, use the New Application wizard. For information about each step of the wizard, see [Creating an application](#).

Note: If you are extending from Pega Foundation for Financial Services Sample application, add PegaFSIF_Sample:AppSetup access group to the new operator record OR if you are extending from Pega Foundation for Financial Services foundation application, add PegaFSIF:AppSetup access group to the new operator. Click the radio button to the left of the access group to select it as the default access group. When this access group is selected as the default access group for an operator, the New Application wizard opens when the operator logs in.

For more information on creating a new operator ID for running the New Application wizard, please see [Creating new operator](#).

Creating the framework application and implementation applications

If your application features vary for different parts of your organization, create a framework enterprise application first, and then create an implementation application for each variation that your organization requires.

Note: If you are extending from Pega Foundation for Financial Services Sample application, add PegaFSIF_Sample:AppSetup access group to the new operator record OR if you are extending from Pega Foundation for Financial Services foundation application, add PegaFSIF:AppSetup access group to the new operator. Click the radio button to the left of the access group to select it as the default access group. When this access group is selected as the default access group for an operator, the New Application wizard opens when the operator logs in.

- For information on creating a framework application and on creating an implementation application on top of framework application, see [Creating an application](#).

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

Note: For application implementation to work as expected in the implementation layer, you need to override the datatransform “PegaFS-Data-AppExtension : AppExtension” that contains variables set to PegaFS data classes in the implementation layer and update the VIP and EMP roles with the implementation layer specific access roles.

Results of running the New Application wizard

For more information on the structure and rulesets generated by the wizard, see [Understanding the classstructure and rulesets generated by the Application Accelerator](#).

Loading sample data

Pega Foundation for Financial Services provides sample data to support the Pega-provided demonstration application and cases. It is recommended that client-supplied sample data be loaded so that the gap analysis efforts are more meaningful for the client.

For more information, see “Importing sample data” in the *Pega Foundation for Financial Services Installation Guide* on the Pega Foundation for Financial Services [product page](#).

Defining the data model

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

Features of Pega Foundation for Financial Services data model include:

- Base data model representing commonly used financial services objects such as individuals, organizations, accounts, and transactions.
- Cross references that allow modelling of complex party-to-party relationships between Organizations and Individuals.
- Customizable integration layer that lets you quickly and seamlessly map the data model to its existing proprietary databases and data services.
- Interfaces that allow standard CRUD (create, read, update, delete) operations on the data.

- A collection of commonly used financial services flow actions are supplied to manage processes associated with financial services objects such as customers, organizations, and accounts.

For information on defining the data model, please see Pega Foundation for Financial Services [product page](#).

- **Pre-configured data types**
- **Mapping the application data**

Pre-configured data types

The Pega Foundation for Financial Services provides a comprehensive financial services data model with many pre- configured Data Types and Data Pages for commonly used data objects that can be leveraged in your application.

The 3 types of data are included:

- **Sample**

Represents Customer and Accounts data to simulate implementor's system of record for testing and demo purposes. Changes frequently over time.

- **Reference**

Used to classify and categorize party (customer) data. Data is static or changes slowly over time.

- **Application Transaction**

Used to drive the flow and events within a case to completion. Data changes frequently over time.

Data Classes:

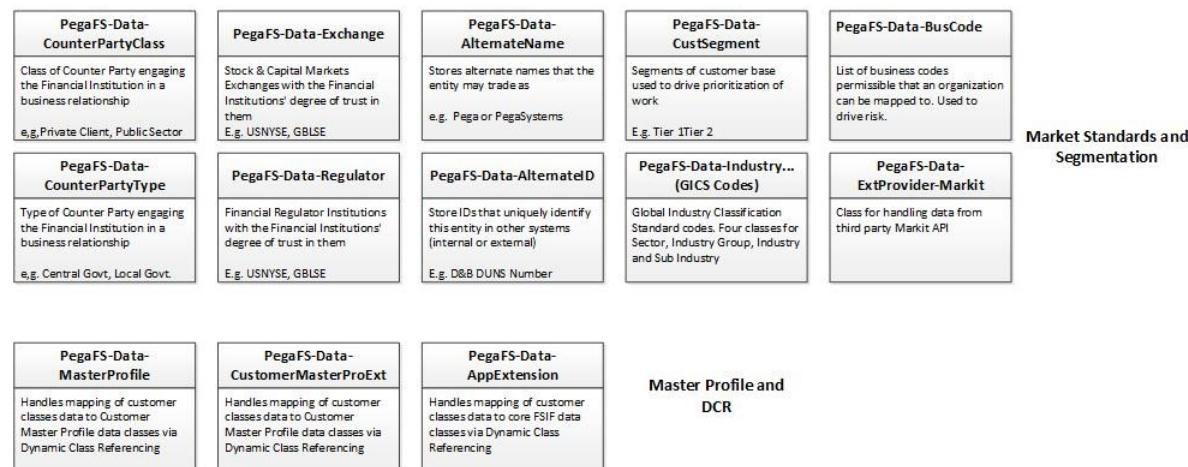
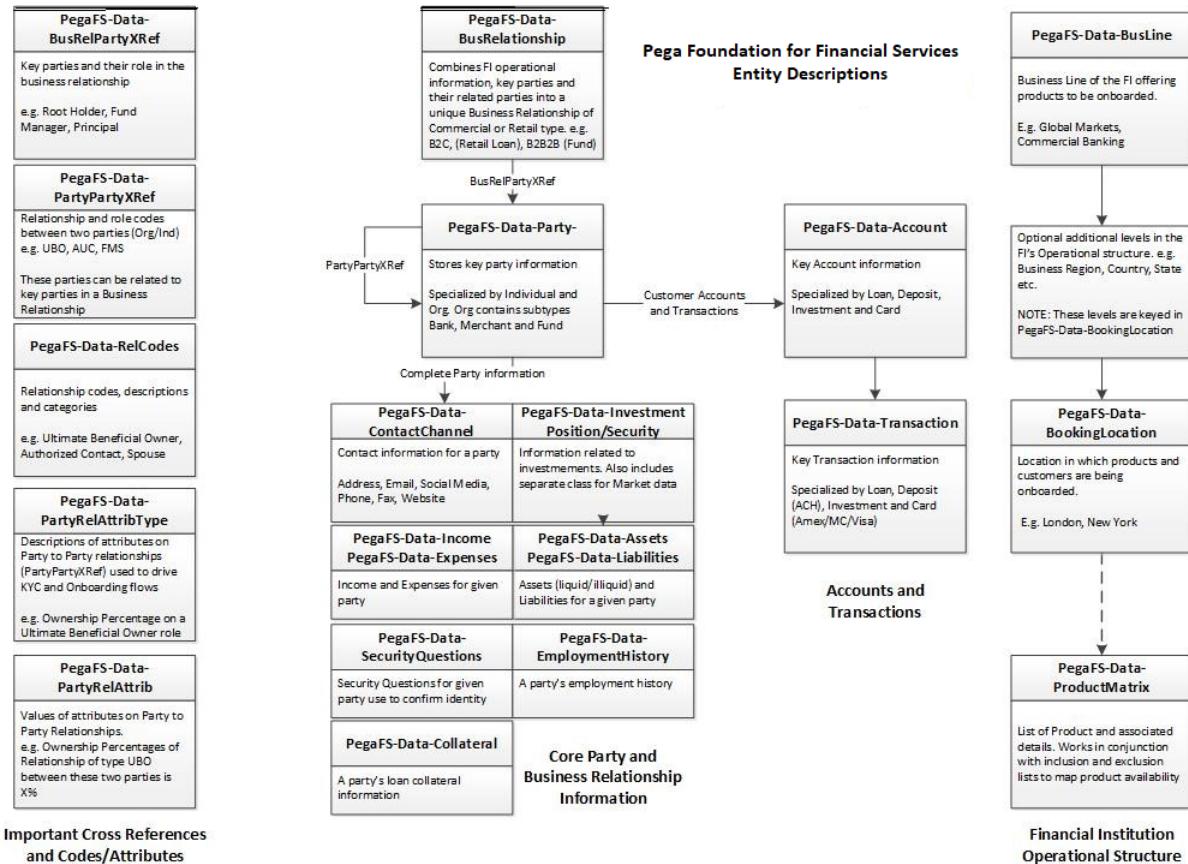
Pega Foundation for Financial Services data classes are located under PegaFS-Data.

Inheritance: PegaFS-Data- ([PegaFS-Data-](#))

	Name	Label	Inheritance type
1	PegaFS-Data-	PegaFS-Data-	Pattern
2	Data-	Data- classes	Pattern
3	@baseclass	@baseclass	NA

[Close](#)

The following diagram highlights the key data types:



For additional information on all available data types and entity relationships, refer to the *PegaFoundation for Financial Services Data Dictionary* on the Pega Foundation for Financial Services [product page](#).

Mapping the application data

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

- Log in to App Studio as a manager.
- In the navigation pane of App Studio, click Data > Data model.
- Click a case type to view the fields for that case type.
- Review the fields and determine which fields to modify, delete, or add.
- To modify an existing field, point to the field, and then click Edit.
- On the Data model tab, determine the properties to add, remove, or leave unchanged.
- Repeat these steps for any other data types, as necessary.

Access groups

Pega Foundation for Financial Services uses access groups that define a set of permissions within an application. Access groups contain permissions for operators (user, manager, administrator), and background processes (event processing, data change tracking).

For more details please refer to [Learning about access groups](#).

Pega Foundation for Financial Services includes operators and access groups for testing and demo purposes installed with PFFS Sample application. These allow you to access the Designer Portal to view and configure the underlying processes and rules, or to launch business processes from a variety of user roles including managers and users. It also includes sample operator IDs for accounting administrators and managers. The password for all operators is **install**. All operators are disabled, use default operator for Pega platform administrator@pega.com to access the system.

Operator ID	Access group	Portal rule
FSIFSYSADMIN	PegaFSIF_Sample	Developer (with access to App Studio)
FSIFSampleSysAdmin	PegaFSIF_Sample	Developer (with access to App Studio)
EventDrivenAgent	PegaFSIF	Developer
FSIFUser	FSIFUser	User
FSIFManager	FSIFManager	Manager
FSIFBusinessAnalyst	FSIFBusAnalyst	Designer Studio (with access to the Requirements Portal)
AccountingSysAdmin	PegaAccounting_SysAdmin	Developer (with access to Accounting rules)
AccountingManager	PegaAccounting_Manager	Accounting manager
AccountingUser	PegaAccounting_User	Accounting user
AccountingVerify	PegaAccounting_SysAdmin	Developer (with access to Accounting rules)
AccountingVerify2	PegaAccounting_SysAdmin	Developer (with access to Accounting rules)
AccountingVerify3	PegaAccounting_SysAdmin	Developer (with access to Accounting rules)
Julian.Weekes@fsi.com	FSIFUser	Case Worker
Jeff.Reynolds@fsi.com	FSIFManager	Case Manager
Akshay.Davis@fsi.com	FSIFBusAnalyst	Designer Studio (with access to the Requirements Portal)

You can use the following operator ID to access the foundation as a system administrator. Use this administrator ID to access and work with the foundation rules

and processes. Account has been disabled by default. Enable the id and set password prior to using:

URL:	http://hostname:port/prweb
Username:	FSIFSysadmin

- **Defining access roles and privileges**
- **Defining your access groups**
- **Configuring access groups**
- **Adding an access group**
- **Adding an access role**

Defining access roles and privileges

You can associate one or more roles to 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.

- Determine which roles are needed for your application. You can use the PFFS roles as a starting point.
- Determine which privileges to associate with each role.
- Associate each role with an access group.

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

Defining your access groups

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

Configuring access groups

A new application includes out-of-the-box access groups and access roles that you can view in App Studio by clicking Users Roles. To associate an access group with a role, follow these steps.

- Log into your application by using the administrator credentials.
- In the navigation pane of App Studio, click **Users Roles**.
- Click **Add role** and enter the details.
- Click **Done**. For more information, see [Access groups](#).

Adding an access group

You might need to add an access group to meet your business needs.

- In the navigation panel of App Studio, click **Users Roles** to display a list of all access groups and access roles that are provided by the application.
- Select the access group that you want to modify and update the details. You can only change the portal details for the access group.
- Click **Done**. For more information, see [Access groups](#).

Adding an access role

You can update or add a new access role only from Dev Studio.

- Log in to Dev Studio by using an administrator account.
- In the navigation pane of Dev Studio, click **Records Security Access Role Name**.
- To update a role, do the following actions:
 - On the **Access Role Name** tab, in the **Access Role** column, click the **Filter** icon.
 - In the **Search Text** field, enter a search term, and then click **Apply**.
 - Click the role that you want to update.
 - On the *role name* tab, update the role details, and then click **Save**.
- To add a new role, do the following actions:

- On the Access Role Name tab, click Create.
- On the New tab, enter a label for the rule configuration record, and configure other settings as needed.
- Click Create and open.
- Update the new role details, and then click Save.

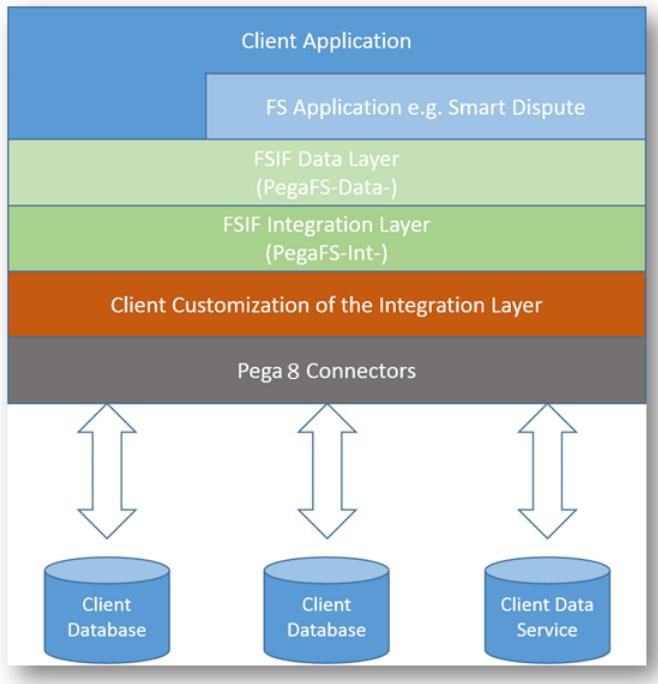
Connecting system of record

- **Integration layer (PegaFS-Int-)**
- **Connect FS products to system of record**

Integration layer (PegaFS-Int-)

Applications built on Pega Foundation for Financial Services interface with and manipulate the classes defined in PegaFS-Data-. The use of these classes lets developers write applications that use the PFFS data model while remaining independent of the underlying physical representation of the data. This is important for the rules delivered with PFFS since the foundation has knowledge of how and to what the classes in the data model will be mapped. When PFFS is installed, one of the first tasks is to connect it to the appropriate backend systems. The PFFS integration layer makes this possible with minimal disruption. Integration classes and rules are in PegaFS-Int- and contained in the PegaFSInt ruleset.

The following graphic shows how the Pega Foundation for Financial Services integration layer fits into the overall application design.



The application avoids a direct dependency on the underlying data stores by building on one or more optional Financial Service (FS) products as well as the PegaFS-Data-classes. In the example above, the FS application is also built on Pega Foundation for Financial Services. This allows for changes in mapping and storage with minimal impact to the top-level application.

This guide presents a high-level overview of the structure of the rules that need to change during the process of mapping Pega Foundation for Financial Services to your system of record.

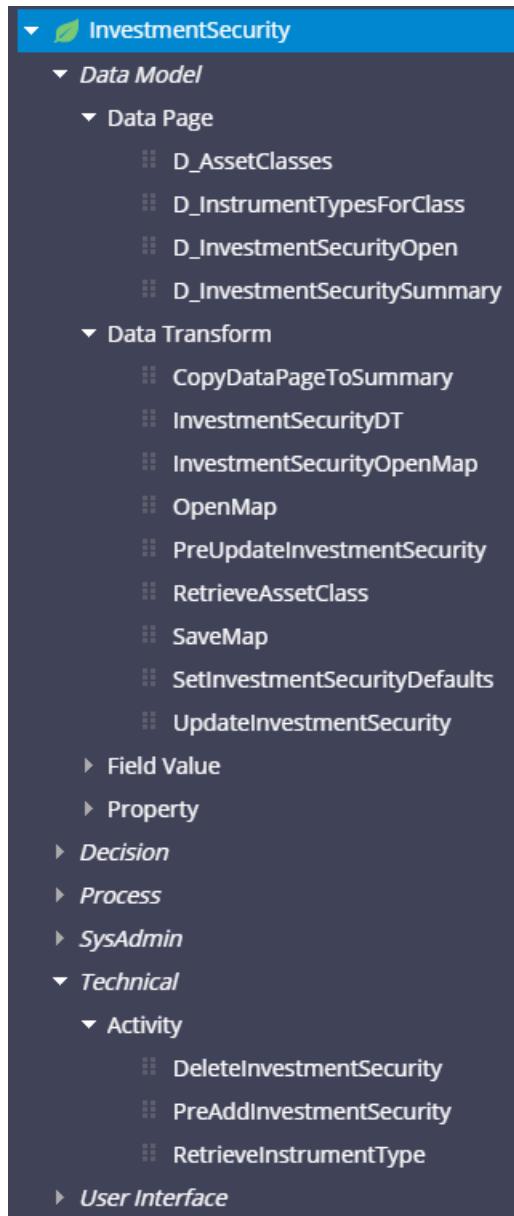
- **Example: Investment security integration**

Example: Investment security integration

An investment security contains information related to a security, for example, a company stock. A security has properties such as security name, issue date, and the exchange on which it is traded.

This section describes interaction between **PegaFS-Data-InvestmentSecurity** data class and corresponding integration class **PegaFS-Int-FSF_SAMPLE_SECURITY** and underlying sample table **FSF_SAMPLE_SECURITY**.

At the Data Layer, the **PegaFS-Data-InvestmentSecurity** class has been defined to be used by the application to store, retrieve, and manage investment securities.



The InvestmentSecurity class contains the properties you would expect when modeling a security. It also defines activities for performing the standard Create, Read, Update and Delete (CRUD) operations on the objects through activities.

Standardized data pages exist that allow for easy access to lists of InvestmentSecurities as well as individual InvestmentSecurity information. Data pages enable developers to easily load, cache, and refresh data. Sections and controls have been updated to accept data pages directly, making it easier to display data.

The consumer of this class is unaware of the details of how these properties are mapped to backend systems. If the application developer builds an application using the properties and activities defined in this class, some or all the backend data mappings can change without impact.

The details about how an InvestmentSecurity is mapped to the underlying data storage are contained in the **PegaFS-Int-FSF_SAMPLE_SECURITY** class. The left graphic below shows the **PegaFS-Data-InvestmentSecurity** class and the right graphic shows the corresponding **FSF_SAMPLE_SECURITY** database table as currently shipped with Pega Foundation for Financial Services 8.6. Note: Sample tables are optionally installed.

Property		column_name
::: AccountName		s_id
::: AssetClass		s_issuer
▶ ::: CurrentPrice		s_assetclass
::: CUSIP		s_instrumenttype
::: Exchange		s_name
::: FaceValue		s_cusip
::: ID		s_isin
::: IndustrySector		s_sedol
::: InstrumentType		s_ticker
::: InterestAccrued		s_reutersid
::: ISIN		s_exchange
::: IssueDate		s_localcurrency
::: Issuer		s_issuedate
::: LocalCurrency		s_industrysector
::: Name		s_parvalue
::: ParValue		s_facevalue
::: ReutersID		s_account
::: SEDOL		s_tenure
::: Shares		s_source
::: StockPrice		s_region
::: Tenure		

The property names defined in the integration class mirror the names of the associated database columns and the class name is like the name of the database. This is not strictly necessary, but it is done for the sake of convenience and clarity. Developers, dealing with the database portion of the interface, typically prefer to use consistent names while developing; it can make debugging easier if the names match.

In an actual implementation, data for the properties defined in the `InvestmentSecurity` class could be stored in more than one data store in the backend system. The default implementations provided with PFFS are pretty much one-to-one with the database for the sake of simplicity. The tables are prefixed with `FSF_SAMPLE_` to indicate that they contain sample data and are meant to be replaced.

Mapping the data:

Data needs to be mapped between the **PegaFS-Data-InvestmentSecurity** (Data) class and the **PegaFS-Int-FSF_SAMPLE_SECURITY** (Integration) class. Three Data Transform rules are defined in **PegaFS-Int-FSF_SAMPLE_SECURITY** to manage these mappings:

- pyDefault – This data transform simply initializes **FSF_SAMPLE_SECURITY** page with default values, usually empty values.
- OpenMap – Data transform called to map data from an integration class page (**FSF_SAMPLE_SECURITY**) to a data class page (**InvestmentSecurity**). This is typically done after data has been loaded from the database into an integration page. The mappings are straightforward but can contain logic to handle situations such as derived values or the mapping of constant values between the database and the application.
- SaveMap – Data transform that maps data in the opposite direction from OpenMap, from a data class page to an integration class page. It is usually called right before data is to be written back to the database, and it could contain logic like that described above for OpenMap.

CRUD operations

Activities need to be implemented to handle the transfer of data to and from the database. The activities in the Data layer call these operations with the data transforms described above to move the data between the Data and Integration layers as well as in and out of the database. Activities defined in PegaFS-Int-FSF_SAMPLE_SECURITY are used to accomplish the data transfer.

Also, a generic activity for CRUD has been created and can be used by implementer:

Class: **@baseclass**

Activity: **PerformCRUD**

Activity: Perform CRUD Operations [Available, Extension]
CL: @baseclass **ID: PerformCRUD** **RS: PegaFS:08-06-01**

Steps **Parameters** Pages & Classes Security Test cases Specifications History

Parameters

Name	Description	Data type	Required	In/Out
1 MapFromPage	Data Layer Page	Page Name	Yes	In
2 MapToPage	Integration Layer Page	Page Name	Yes	In
3 CRUD	1 for Create 3 for update and 4 for delete	Integer	Yes	In
4 MapFromPageClass	Data Layer Page Class	String	Yes	In
5 MapToPageClass	Integration Layer Page Class	String	Yes	In
6 MappingDataTransf	Mapping Data Transform (Data-Integrati)	String	Yes	In
7 ValidateRuleName	Validate Rule Name	String	Yes	In

Data pages:

Every data class in PFFS carries at least 2 standardized data pages that return:

- List of records: D_InvestmentSecuritySummary calls report definition in PegaFS-Int-FSF_SAMPLE_SECURITY.

Data Page: List of investment securities [Available]
ID: D_InvestmentSecuritySummary **RS: PegaFS:08-06-01**

Save as Actions Private edit

Definition Load Management Parameters Pages & Classes Test cases Usage Specifications History

Data page definition

Structure: List
Object type*: PegaFS-Data-InvestmentSecurity
Mode: Editable
 Enable lightweight clipboard mode
Scope: Thread

Data Sources

If there are multiple sources, the first IF that evaluates to true is used. If none evaluate to true, the OTHERWISE source is used.

Source *	Name *	Response Data Transform *
Report definition	GetAllFSFSampleSecurities	InvestmentSecuritySum
PegaFS-Int-FSF_SAMPLE_SECURITY	Parameters	Parameters

Simulate data source System name: ClientReferenceSystem

- Lookup a record: D_InvestmentSecurityOpen retrieve a single record as a Page based on primary key(s) from PegaFS-Int-FSF_SAMPLE_SECURITY

Data Page: Lookup an investment security details [Available]
ID: D_InvestmentSecurityOpen RS: PegaFS:08-06-01

Definition Load Management Parameters Pages & Classes Test cases Usage Specifications History

Data page definition

Structure
Page

Object type*
PegaFS-Data-InvestmentSecurity

Mode
Editable

Enable lightweight clipboard mode

Scope
Thread

Data Sources

If there are multiple sources, the first IF that evaluates to true is used. If none evaluate to true, the OTHERWISE source is used.

IF IsDPPParameterEmpty

Source* Data Transform Data transform name* SetEmptyResultsForData

ELSE IF isNotNull

Source* Data Transform Data transform name* InvestmentSecurityDT

OTHERWISE

Source* Lookup Class name* PegaFS-Int-FSF_SAMPLE_SECURITY

Response Data Transform* InvestmentSecurityOpen

Run response data transform on error

DISABLED DATA SOURCE DETAILS
Lookup • PegaFS-Int-FSF_SAMPLE_SECURITY (PegaFS-Int-FSF_SAMPLE_SECURITY)

Connect FS products to system of record

Financial Services (FS) products can be created using Product designer for Financial Services (PDFS), or manually. Users of FS products run the following FS applications:

- Pega Marketing for Financial Services (PMFS) - Ingests FS products from PDFS application and creates marketing offers that can be executed.
- Sales Automation for Financial Services (SAFS) - FS Products are used in the creation of Opportunities.
- Client Lifecycle Management for Financial Services (CLM) - FS Products are used in product selection of onboarding process.

- Pega Know Your Customer (KYC) - FS Products are used to perform extra risk due diligence where appropriate.
- **Before connecting to another system of record**
- **Data storage and structure of products**
- **Class structure and rules**
- **Connecting to another data source**
- **References**

Before connecting to another system of record

Pega Foundation for Financial Services (PFFS) requires Operating and Product structures to be set up before creating/adding products. This is essential in the product selection process for FS applications.

- **Configuring the operating structure**
- **Configuring the product structure**

Configuring the operating structure

Follow these steps to configure the operating structure:

1. In Dev Studio, click **Configure > Financial Service > Operating Structure**
2. Set up the operating structure as shown in the sample below:

Operating Structure						
View operating structure for UPlus Financial Services						
Organizational chart		Taxonomy				
Entity name	Level	Calendar	Currency	Departments	Financial business segment	
UPlus Financial Services	L+ Add Financial Institution	UPFSDefault	USD	0		
Asset Management	L+ Add Business Line	UPFSDefault	USD	8	Corporate & Investment Banking	
Global Business Banking	L+ Add Business Line	UPFSDefault	USD	0	SMB	
Global Consumer Banking	L+ Add Business Line	UPFSDefault	USD	7	Retail	
Austria	L+ Add Country	UPFSDefault	USD	0	Retail	
Vienna	L+ Add City	UPFSDefault	USD	0	Retail	
Vienna Central Branch	Branch	UPFSDefault	USD	0	Retail	
Australia	L+ Add Country	UPFSDefault	USD	0	Retail	

Operating structure

Configuring the product structure

Follow these steps to configure the product structure:

1. In Dev Studio, click **Configure > Financial Service > Product Structure**
2. Set up the product structure as shown in the sample below:

Product Structure						
View product structure for UPlus FS Product Operations						
Product Chart		Taxonomy				
Entity name	Level	Calendar	Currency	Departments	Financial business segment	
UPlus FS Product Operations	L+ Add Financial Institution	USDefault	USD	0		
Product Operations	L+ Add Business Line	USDefault	USD	1		
Corporate and Investment Banking	L+ Add Segment	USDefault	USD	1		
Global Corporate Finance	L+ Add Line of business	USDefault	USD	1		
Global Corporate Finance	L+ Add Product Category	USDefault	USD	0		
Asset Based Lending	L+ Add Product Type	USDefault	USD	0	Corporate & Investment Banking	
Asset Finance	L+ Add Product Type	USDefault	USD	0	Corporate & Investment Banking	
Debtor Protection	L+ Add Product Type	USDefault	USD	0	Corporate & Investment Banking	
Invoice Finance	L+ Add Product Type	USDefault	USD	0	Corporate & Investment Banking	
Mergers and Acquisitions	L+ Add Product Type	USDefault	USD	0	Corporate & Investment Banking	

Product Structure

- Operating Structure and Product Structure items created or edited in the UX are saved in the Pega system PR_DATA_ADMIN table.
- **UPFS** is the financial institution abbreviation representing the Operating (Organization) Structure.
- **UPFSPD** is the abbreviation that represents the Product Structure.
- The path listed in the **PYORGUNIT** column will be matched with data in PFFS tables to determine products availability in product selection.

Note: The format of the **PYORGUNIT** data shown below needs to match the data path in the system of record for the product selection process to work:

pyname	pyorgdivision	pyorgunit	pyorganization
BratislavaWealthOffice	PWM	UPFS-PWM-SK-BR-BROff	UPFS
Boston	PWM	UPFS-PWM-US-Boston	UPFS
BeaconHillOffice	PWM	UPFS-PWM-US-Boston-BHOff	UPFS
NationalOffice	PWM	UPFS-PWM-US-Boston-NatOff	UPFS
NewYorkCity	PWM	UPFS-PWM-US-NYC	UPFS
CentralParkOffice	PWM	UPFS-PWM-US-NYC-CPOff	UPFS
Cards	PDOPB	UPFSPD-PDOPB-RET-CARDS	UPFSPD
AssetBasedLending	PDOP	UPFSPD-PDOP-CIB-GCF-GCF-ABL	UPFSPD
AssetFinance	PDOP	UPFSPD-PDOP-CIB-GCF-GCF-AF	UPFSPD
SydneyOperaOffice	PWM	UPFS-PWM-AU-Sydney-SOOff	UPFS

Data Admin table

Data storage and structure of products

Pega Foundation for Financial Services (PFFS) hosts and manages products in terms of data structure and storage.

Storage: Product data and availability is stored in the following PFFS tables:

- FSF_PRODUCTMATRIX
- FSF_PMINCLUSIONXREF
- FSF_PMEEXCLUSIONXREF

The FSF_PRODUCTMATRIX table contains basic information records on products, services and product bundles. Each product record belongs to a product category and a line of business. Columns in the table are defined as name/value pairs. (for example, PM_VAR[No]_NAME/PM_VAR[No]_VALUE). The columns that need to be populated are **PM_PRODUCTNAME**, **PM_PRODUCTID**, **PM_VAR1_NAME/PM_VAR1_VALUE** for Line of Business, and **PM_VAR2_NAME/PM_VAR2_VALUE** for Product Category.

- Sample products in FSF_PRODUCTMATRIX table

Sample products in FSF_PRODUCTMATRIX table

pm_productname	pm_productid	pm_var1_name	pm_var1_value	pm_var2_name	pm_var2_value
Low-risk Packaged Mortgages	P-126	Line of Business	GlobalMarkets	Product Category	FixedIncome
SBA 504 Loan	P-100	Line of Business	Lending	Product Category	Secured
SBA 7(a) Loan	P-101	Line of Business	Lending	Product Category	Secured
SBA Express Loan	P-102	Line of Business	Lending	Product Category	Secured
Online Wire Transfer	P-103	Line of Business	Services	Product Category	Online
US Government Repo 5 years	P-127	Line of Business	GlobalMarkets	Product Category	FixedIncome
Remote Deposit Capture	P-104	Line of Business	Services	Product Category	Online
New Auto Loan	P-105	Line of Business	Lending	Product Category	Secured
Personal Loan	P-106	Line of Business	Lending	Product Category	Unsecured

Product matrix

FSF_PMINCLUSIONXREF is a cross reference table that contains line of business indicating that all product records for the given line of business are allowed.

Sample product lines of business, operating structure paths in pi_inclusionpath, and product structure paths in the pi_marketsegmentpath columns:

pi_id	pi_productlobvalue	pi_inclusionpath	pi_marketsegmentpath
100 (null)		UPFS-GCB-US	UPFSPD-PDOP-RET
101 (null)		UPFS-GBB-US-NYC	UPFSPD-PDOP-SMB
102 (null)		UPFS-GBB-US-Boston	UPFSPD-PDOP-SMB
103 (null)		UPFS-GBB-US-NYC	UPFSPD-PDOP-SMB
104 (null)		UPFS-GCB-US-Boston	UPFSPD-PDOP-RET
105 (null)		UPFS-GCB-US-NYC	UPFSPD-PDOP-RET
106 (null)		UPFS-PWM-US-Boston	UPFSPD-PDOP-WM
107 (null)		UPFS-PWM-US-NYC	UPFSPD-PDOP-WM
108 Global Markets		UPFS-GM	UPFSPD-PDOP-CIB
109 PrimeBrokerage		UPFS-GM	UPFSPD-PDOP-CIB

Product Matrix Inclusion Table

The FSF_PMEXCLUSIONXREF table contains records that block specific product ids or all products belonging to a product category from being allowed.

Sample operating structure paths in the pe_exclusionpath column.

pe_id	pe_idtype	pe_exclusionpath
Commodities	ProductCategory	UPFS-GM-APAC-SG
Convertible Bonds	ProductCategory	UPFS-GM-AME-CA
Convertible Bonds	ProductCategory	UPFS-GM-APAC-HK
Convertible Bonds	ProductCategory	UPFS-GM-APAC-SG
Derivatives	ProductCategory	UPFS-GM-APAC-HK
Derivatives	ProductCategory	UPFS-GM-EMEA-GB-MSLtd
Equities	ProductCategory	UPFS-GM-AME-CA
Equities	ProductCategory	UPFS-GM-APAC-HK
Equities	ProductCategory	UPFS-GM-EMEA-FR

Product Matrix Exclusion Table

Logic to determine availability retrieves a list of product records from PRODUCT MATRIX for specified lined of business (INCLUSION), and omits from that list products or all products of a specified product category (EXCLUSION).

INCLUSION – EXCLUSION = List of available products from PRODUCT MATRIX.

FSF_PRODUCTMATRIX is populated either by using the PDFS application or manually by using SQL. Data in FSF_PMINCLUSIONXREF and FSF_PMEXCLUSIONXREF can only be populated using SQL.

Class structure and rules

Integration class: *PegaFS-Data-ProductMatrixInclusionXref*

Report Definition:

- *GetLOBByLocation* takes Organizational structure path of operator and matches with inclusion path.

Integration class: *PegaFS-Data-ProductMatrixExclusionXref*

Report Definitions:

- *GetExclusionsByProductId* takes Organizational structure path of operator and searches for type: product to return list of product ids.
- *GetExclusionsByPMVar2Value* takes Organizational structure path of operator and searches type: product category to return list of product categories.

Integration class: PegaFS-Int-FSF_PRODUCTMATRIX

Report Definitions:

- *FilterProductsByID*
- *GetAllowedProductsByLOB*
- *GetAllowedProductsByLocation*
- *GetDepositProductsByType*
- *GetExclusionsByProductId*
- *GetExclusionByVar2Value* where Var2 is product category.
- *GetProductFromProductMatrix*
- *LOBList*
- *ProductCategoryList*
- *ProductMatrixSummary*
- *ProductMatrixSummary_Short* returns fewer columns than version above.

Data Class: PegaFS-Data-ProductMatrix

Data Pages:

- *D_AllowedProductByLocation* retrieves available products. Invokes the activity *GetAllowedProductsByLocation*.
- *D_AvailableLOB* retrieves available lines of businesses from inclusion.
- *D_AvailableProductCategory* retrieves list of product categories not listed in exclusion.
- *D_GetAllowedProductsByLocation* retrieves list of products by location.
- *D_GetProductFromProductMatrix*

- *D_LOBList* retrieves lines of business.
- *D_ProductApplicationsByType*
- *D_ProductCategoryList*
- *D_ProductMatrixSummary* returns a list of all products and bundles with details.
- *D_ProductOpenById* lookup that returns a single complete record from Product Matrix.

Activity:

GetAllowedProductsByLocation applies formulas based on input parameters:

- INCLUSION – EXCLUSION = AVAILABLE LIST OF PRODUCTS.
- INCLUSION = AVAILABLE LIST OF PRODUCTS (when products are not excluded).

Connecting to another data source

When choosing to connect to another data source for FS products, the implementer must consider the following changes to the classes and rules listed above.

Data Sources

- Another table(s)
 - New table(s) will be imported into Pega via a wizard with integration classes and properties representing columns created automatically as a result.
 - Every report definition listed above will be replaced with new report definitions hosted by newly created integration classes.
 - Data classes can be re-used with overrides to data pages and mapping data transforms in the implementation layer.
 - Every data page listed above will be overridden in the implementation layer to point to new report definitions.
 - Data transforms responsible for mapping are carry specific naming convention: OPENMAP for mapping retrieved data and SAVEMAP for mapping data back from class to data source. They can be overridden at the implementation layer with mappings for additional properties.

- GetAllowedProductsByLocation activity will be overridden in the implementation layer to use new report definitions for product selection logic.
- Re-using existing tables
 - ETL job(s) can be setup to synchronize data between FS product tables and other system of record (data sources) platform.
 - Existing integration and data classes can be extended with additional properties to be mapped and columns to be retrieved.
 - Report definitions can be overridden at implementation layer to return new integration class properties (table columns).
 - Data transforms responsible for mapping are carry specific naming convention: OPENMAP for mapping retrieved data and SAVEMAP for mapping data back from class to data source. They can be overridden at the implementation layer with mappings for additional properties.
- Service calls
 - Integration class with connector rules will be created for authentication and using exposed service methods.
 - Data classes can be re-used with overrides to data pages and mapping data transforms in the implementation layer.
 - Every data page listed above will be overridden in the implementation layer to point to connector rules.
 - Data transforms responsible for mapping are carry specific naming convention: OPENMAP for mapping retrieved data and SAVEMAP for mapping data back from class to data source. They can be overridden at the implementation layer with mappings for additional properties.

References

The following are the references for this wiki article:

[Data and Integration: Data Pages](#)

[Integrating Data](#)

Integration Service Connectors

Managing reference data

Pega Foundation for Financial Services provides a reference to maintain your enterprise repository in Pega. You must make an inventory of all reference data that you chose to maintain in Pega Foundation for Financial Services.

- fsf_partyrelattrib
- fsf_partyrelattribxref
- fsf_partyrelatttype
- fsf_partyrelcodexref
- fsf_partytype
- fsf_ref_acctprotypedesc
- fsf_ref_acctypedesc
- fsf_ref_altid
- fsf_ref_altname
- fsf_ref_businesscode

- fsf_ref_businessgoals
- fsf_ref_counterptyclass
- fsf_ref_counterptytype
- fsf_ref_country
- fsf_ref_cp_buscodesxref
- fsf_ref_cp_relcodexref
- fsf_ref_cpclasstypexref
- fsf_ref_cptype_reltypexref
- fsf_ref_currency
- fsf_ref_currencyexchangerate
- fsf_ref_custpotvalue
- fsf_ref_custsegment
- fsf_ref_doccatfamilyxref

- fsf_ref_exchange
- fsf_ref_industry
- fsf_ref_industrygroup
- fsf_ref_lu_marketsegment
- fsf_ref_merchantcategorycodes
- fsf_ref_party_mktsegxref
- fsf_ref_partylocalsubtype
- fsf_ref_partysubtype
- fsf_ref_partysubtypexref
- fsf_ref_provincestate
- fsf_ref_regulator
- fsf_ref_sector
- fsf_ref_subindustry

- fsf_relcodes

Note: The above reference data comes pre-loaded with data. You are required
 ⓘ to update or remove and upload your own reference data using the upload
 capabilities provided in the solution.

Application configurations

- Updating the localization of currency fields

Updating the localization of currency fields

Localizing currency involves using symbols or codes to indicate the currency of the current number. Depending on the context, a special symbol (\$ or US\$), or an ISO 4217 currency code (USD) is used to represent the currency.

To localize the PFFS Properties currency code, the currency of these properties must be configured from System of Records of the respective data.

While mapping the data from Integration layer to Data Layer the Property qualifiers on these properties should be mapped to correct currency as per the locale in the respective class OpenMap data transform.

- AccountBalance
- NetTrxnAmount
- HouseholdIncome
- Assets
- Liabilities
- Expenses
- Revenue

 **For example:**

Updating the AccountBalance property currency as per locale to USD.

Home AccountBalance

Property: Account balance [Available]

Edit input AmountWithCurrency

Use validate AmountWithCurrency

Column inclusion

Persistence

Do not save property data

Security

Cannot be declarative target Declarative network

Cannot be included as input field

Allow use as reference property in activities

Cannot be localized in UI controls

Property qualifiers

Qualifier	Value
pyCurrency	.CurrentBalanceCurrCode

Account balance

Property Qualifier on AccountBalance is CurrentBalanceCurrCode that should be mapped to required locale in the OpenMap data transform of respective class as per locale required from the SOR of the data.

Entry	Source Field	Target Field
3.7	Set .LastTranSeq	.A_LASTTRANSEQ
3.8	Set .AccountStatus	.A_STAT
3.9	Set .AccountStatusDesc	.A_STATUS
3.10	Set .AccountSubStatusDesc	.A_SUBSTATUS
3.11	Set .AccountSubStat	.A_SUBSTAT
3.12	Set .AccountType	"Deposit"
3.13	Set .AccountTypeDesc	.A_ACCTDESC
3.14	Set .CifNbr	.A_CIFNBR
3.15	Set .CurrentBalanceCurrCode	@if(MapFromPage.A_CURBAL_CURR='

OpenMap data transform

If sample database tables are mapped for testing, then the pegadata.fsf_sample_acct table should be updated with USD in a_currbal_cur column. In OpenMap the CurrentBalanceCurrCode will be mapped to a_currbal_cur to display USD on UI.

Integrations

You must inventory which external applications exist at your site and plan how to connect to them. Some common integration points for the Pega Foundation for Financial Services include interfaces to:

External Service Provided	Provider Name
Screening of Organizations	Refinitiv World-Check One
Enrichment of Organizations	Dun & Bradstreet Direct API

External Service Provided	Provider Name
Enrichment of Organizations	IHS Markit
Enrichment of Organizations	Refinitiv AVOX Legal Entity
Enrichment of Organizations	SWIFT KYC
Credit Profile of Individuals	Equifax InterConnect
Authenticate/Verify	eIdentification

Note: Identify integration points as early in your planning as possible. If a connection to an external data source is required and either the data itself does not exist or the interface to that data does not exist, you must account for the time to build the interface to this application.

- [Navigating to the pre-configured external data pages](#)

Navigating to the pre-configured external data pages

1. If you have imported the Pega Foundation for Financial Services demo application and sample operators as part of installation, log in to the application as **Core Administrator** using the password that you specified when you enabled this operator; to enable Pega-provided operators, see [Enabling operators](#). Otherwise, complete the following steps to create a new operator ID:
 - a. Login 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 Administrator operator.
 - c. AddPegaFSIF:Administrators access group to the new operator record. Click the radio button to the left of the access group to select it as the default access group.
 - d. Save the new operator record.

- e. Log in as the new operator that you created.
2. Navigate to Configure > Data Model > View external data entities.
3. Launch App Studio and click Data.

This link allows you to view and configure the data types defined in your app. Data types allow you to define and manage simple data records that you can reference from your case types. It helps application business architects easily identify different systems of record that they will have to connect to, and different data types interfaces they will need to integrate with to start using the application.

Defining the security model and organization structure

- Security model
- Organizational structure
- Defining your authentication scheme

Security model

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. Authentication information is presented in [Defining your authentication scheme](#) part of the document.

Organizational structure

Pega Foundation for Financial Services has a predefined organization, division, and organizational unit for example purposes. During implementation the actual operational structure of the Financial Institution that is using Pega would be defined.

Select the **Configure> Org & Security > Organization > Organizational Chart** landing page option in **Dev Studio** to display the organizational structure.

▼ UPFS	UPPlus Financial Services
▼ AM	Asset Management
UPFS-AM-AM	Asset Management
▼ GM	Global Markets
UPFS-GMs-GMs	General Markets
▼ UPFS-GM-AME	Americas
UPFS-GM-AME-CA	Canada
UPFS-GM-AME-US	United States
▼ UPFS-GM-APAC	APAC
UPFS-GM-APAC-AU	Australia
UPFS-GM-APAC-HK	Hong Kong
UPFS-GM-APAC-SG	Singapore
▼ UPFS-GM-EMEA	EMEA
UPFS-GM-EMEA-FR	France
UPFS-GM-EMEA-GB	Great Britain
UPFS-GM-GM	Global Markets
UPFS-Invest-Invest	Investment
▼ TB	Transactional Banking
UPFS-TB-TB	Transactional Banking

To support more complex Business Relationships, especially in the Commercial and Investment Banking space, the standard Pega Organization and Security functionality has been extended in the foundation. This allows for the ability to create and name additional levels in the operational hierarchy between the required levels of Financial Institution, Business Line and Booking Location.

Select the **Designer Studio > Financial Services > Operational Structure** landing page option to display the organizational structure. A reference Financial Services Institution called UPlus Financial Services is shipped with PFFS to demonstrate the Operational Structure functionality. This ships with the necessary core Workbaskets for routing work.

Operating Structure

View operating structure for [UPlus Financial Services](#) ▾

Entity name	Level	Calendar	Currency	Departments	Financial business segment
UPlus Financial Services	L+ Add Financial Institution	UPFSDefault	USD	0	
Asset Management	L+ Add Business Line	UPFSDefault	USD	8	Corporate & Investment Banking
Global Business Banking	L+ Add Business Line	UPFSDefault	USD	0	SMB
Global Consumer Banking	L+ Add Business Line	UPFSDefault	USD	7	Retail
Global Markets	L+ Add Business Line	UPFSDefault	USD	9	Corporate & Investment Banking
Americas	L+ Add Region	UPFSDefault	USD	6	Corporate & Investment Banking
Canada	L+ Add Country	UPFSDefault	USD	6	Corporate & Investment Banking
UPlus-Ottawa	Booking Entity	UPFSDefault	USD	1	Corporate & Investment Banking
United States	L+ Add Country	UPFSDefault	USD	6	Corporate & Investment Banking
UPlus-New York	Booking Entity	UPFSDefault	USD	1	Corporate & Investment Banking
UPlus-Washington	Booking Entity	UPFSDefault	USD	1	Corporate & Investment Banking
APAC	L+ Add Region	UPFSDefault	USD	6	Corporate & Investment Banking
EMEA	L+ Add Region	UPFSDefault	USD	6	Corporate & Investment Banking
Private Wealth Management	L+ Add Business Line	UPFSDefault	USD	0	Private Wealth Management

Click **Taxonomy** for an additional view of the UPlus Financial Services operational structure.

Operating Structure

View operating structure for **UPlus Financial Services**

Organizational chart [Taxonomy](#)

All Financial Institutions should contain three default levels: financial institution, business line, and a renameable bottom level.

Operational level	Level type	Access groups
▼ Financial Institution	L+ Add	Business Yes
▼ Business Line	L+ Specialize L+ Add	Business No
▼ Country	L+ Add	Jurisdictional No
Branch		BalanceSheet No
▼ Business Line (CIB)	L+ Add	Business No
▼ Region	L+ Add	Geographical No
▼ Country	L+ Add	Jurisdictional No
Booking Entity		BalanceSheet No
► Business Line (Retail)	L+ Add	Business No
► Business Line (SMB)	L+ Add	Business No
► Business Line (Wealth)	L+ Add	Business No

Click on the **New** button on the top right of the screen to create a new operational structure for the organization. Specify the Organization details such as name, default calendar, and currency as well as the Access Group information. Click **Submit** to create the new organization.

New Financial Institution

Name *	My Bank	Access groups
ID *	MyBank	User Profile Access group ID
Top level class *	Data-Admin	System Administrator PegaFSIF
Calendar *	USDefault	Business Analyst FSIFBusAnalyst
Currency *	USD	Team Manager FSIFManager
		Operator FSIFUser
		Add user profile

Cancel **Submit**

The new Financial Institution is created with an operational structure containing three organizational levels by default as shown below:

Operational Structure

View operational structure for **My Bank**

[Organizational chart](#) **Taxonomy**

All Financial Institutions should contain three default levels: financial institution, business line, and a renameable bottom level.

Operational level	Level type	Access Groups
Financial Institution	Business	Yes
Business Line	Business	No
Branch	Geographical	No

[Add level](#)

Additional operational levels can be added to the structure by clicking the **Add level** button.

Add operational level X

Name *	Region	Access groups (i)
Parent level	Business Line	User Profile Access group ID
Level type	Geographical	Business Analyst MyBank_Region_FSIFBusAnalyst (i)
+ Add user profile		
Cancel		Submit

You can create a default workgroup and workbasket for each of the additional business levels by using the **Add user profile** link.

There is typically one Financial Institution defined per installation. Once this is defined, you can populate nodes in the operational hierarchy by clicking **Add link** and entering details for each of the levels.

The screenshot shows the Pega Platform's operational structure management interface. A modal window titled "Add Business Line" is displayed, containing fields for Name, ID, Financial Business Segment, Calendar, and Currency. The "Name" field is populated with "MyBank". The "ID" field has an error message: "Value cannot be blank". The "Financial Business Segment" field is set to "Corporate & Investment Banking". The "Calendar" field is set to "USDefault". The "Currency" field is set to "USD". At the bottom right of the modal is an orange "Save" button.

For information on defining security model and organizational structure, see [LINK](#)

Defining your authentication scheme

- [Authentication schemes](#)
- [Implementing your authentication scheme](#)
- [Authorization scheme](#)

Authentication schemes

The Pega Platform offers the following authentication types:

- PRBasic

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

- PRSecuredBasic

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

- PRCustom

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

- PRExtAssign

Supports external assignments (Directed Web Access).

- J2EEContext

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

Implementing your authentication scheme

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

- Discuss the authentication schemes with your site's security and application server teams.
- Determine the appropriate authentication type.

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

Authorization scheme

Pega Foundation for Financial Services comes with a predefined set of access groups, roles, and privileges as described in the [Defining Your Access Groups](#) part of this document. 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.

Customizing the user experience

While Pega Foundation for Financial Services is fully functional immediately after installation, you might change portions of the user experience (UX) to meet the needs of the users at your site.

- Reviewing content of the portals
- Review the content of your application's portals to ensure that only relevant data is presented. PFFS provides out of the box portals:
- Accounting manager
 - Accounting user
 - Requirements
- Designing for screen performance.

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

- Review transactional service level agreements.
- Review service performance, such as how long it takes back end systems to gather the data needed to display customer details.
- Review data requirements to determine which fields contain the information that the representatives absolutely need for the task that they need to perform.
- Determine if any network configuration could cause latency.

Features implementation

Pega Foundation for Financial Services provides a rich set of features designed to enhance the customer experience, improve user productivity and increase customer satisfaction.

This section describes key capabilities and features of the application that you can use as-is or extend to meet your business needs.

- **Financial Services core features**
- **T.O.M - Reusability of org charts created by other methods**
- **Requirements management**
- **Accounting**
- **Third party data providers**
- **Data management**
- **Externalization of third party data**
- **Adding a new data source for entity enrichment**
- **Currency conversion structure**
- **Soft delete functionality**
- **Operating structure management (T.O.M)**
- **Reusability of existing org charts to operating structure**
- **Configuring data for product availability processing in PFFS**
- **Creating a sample financial product**
- **Using an enrichment or screening marketplace component**

Financial Services core features

- Base data model representing common financial services objects such as individuals, organizations, accounts and transactions.
- Cross references allow modelling of complex party-to-party relationships between Organizations and Individuals.

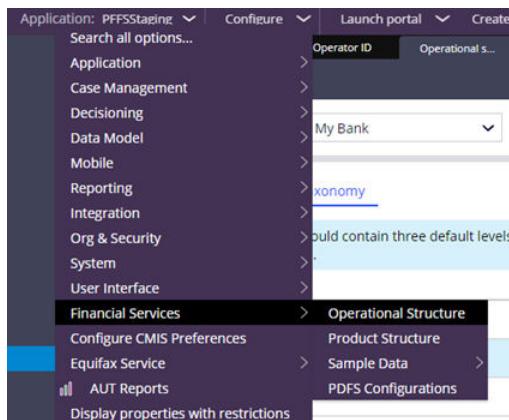
- Customizable integration layer that lets you quickly and seamlessly map the data model to its existing proprietary databases and data services.
- Interfaces allow standard CRUD (create, read, update, delete) operations on the data.
- A collection of commonly used financial services flow actions are supplied to manage processes associated with financial services objects such as customers, organizations, and accounts.
- Event driven service to launch case processes. For more information please see [Event Driven Architecture](#).
- Change tracker (audit) service for recording value changes in monitored properties.



Note: To refer to the event driven architecture of previous releases, see [Event Driven Architecture 1g](#).

T.O.M - Reusability of org charts created by other methods

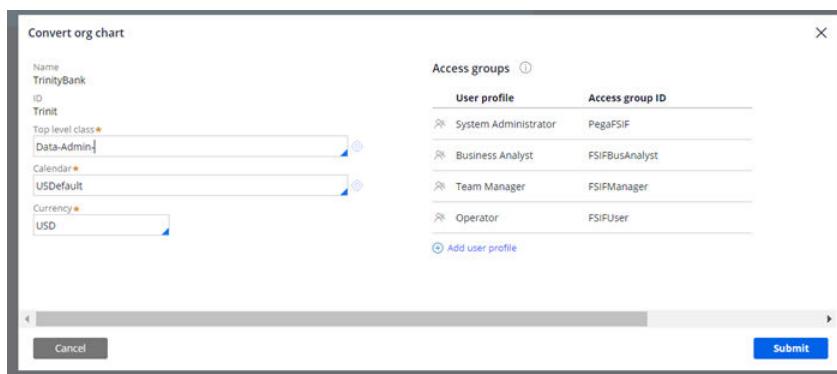
- Organizational charts created either through Organizational Chart Landing Page or New Application Wizard can be converted into Operating Structure through 'Convert' button.
- Navigate to Operating Structure Landing Page through Configure à Financial Services à Operational Structure.



- Select the Org chart (that you wanted to convert) from “View Operating Structure” dropdown and click on ‘Convert’ Button.



- Convert org chart' modal pops up on the screen with prepopulated data for Name and ID. Enter input for fields Top level class, Calendar and Currency and click Submit.



- The organizational chart is converted and initiated with required fields like NodeID, OrgLevel and the associated taxonomy can be seen in Operating Structure LP.

Requirements management

- Enables you to upload supporting documents required by the system.
- Provides an interface for you to review, manage, and approve user documents.

For more information on requirements please see Requirements guide on Pega Foundation for Financial Services [product page](#).

Accounting

- Offers a robust set of sub-ledger accounting and adjustments facilities. Supports end-to-end management of financial transactions, including suspense and

adjustment accounting, write-off management, accounting cut-off management, multi-level verification, and correspondence generation.

For more information on accounting please see Accounting guide on Pega Foundation for Financial Services [product page](#).

Third party data providers

- Optional components for enriching company data from financial information providers enables financial services organizations many benefits including: operational efficiencies, more efficiently managing credit decisions, verifying identity, increasing fraud awareness, providing more targeted and personalized cross-sell and upsell strategies, and optimizing collections treatments.
- Pega's ecosystem of pre-built connections to providers are available on Pega Marketplace and can be incorporated into Pega Foundation for Financial Services using a centralized set of objects.
- The following are the pre-built connectors currently available from Pega Marketplace and have pre-built configurations with the Foundation for Financial Services centralized Business Service.

Service Provided	Provider Name
Screening of Organizations	Refinitiv World-Check One
Enrichment of Organizations	Dun & Bradstreet Direct API
Enrichment of Organizations	IHS Markit
Enrichment of Organizations	Refinitiv AVOX Legal Entity
Enrichment of Organizations	SWIFT KYC
Credit Profile of Individuals	Equifax InterConnect
Authenticate/Verify	eldentification
Credit Report of consumer	Equifax

For more information on 3rd party providers, please see components and documentation on [Pega Marketplace](#).

OneView Equifax Component

Delivers the Consumer Credit File along with Alternative Data assets via a single inquiry. The OneView solution helps businesses make faster, well-informed decisions and assess credit risk with access to a holistic view of a consumer via a single delivery to improve decision making with optional scoring models and fraud protection tools.

OneView component is built on Pega Infinity which can be implemented in any prior version above 8.5 or above. As part of Pega Foundation for Financial Services, A Sample case type 'Customer credit profile' leveraging the OneView is implemented to test OneView API.

The Case Type will be comprised of Customer search screen (from internal Sample tables), Post selecting the customer the Credit profile for the selected customer is retrieved using Equifax OneView. Request will be formed using the Customer details like Name ,SSN. Response will have the details of Credit score, Reasons for credit score, Bankruptcy details, 3rd party collection details, Trades, Address details and Employment details which will be displayed on customer Credit profile information.

Data management

Pega Foundation for Financial Services hosts data management utilities for change tracker data (audit) backup, sample and reference data backups (data snapshots) and date shifting of sample accounts and transactions for demo and testing purposes. Utilities are accessible under **Configure > Financial Services** in **Dev Studio**.

Change Tracker (Audit): Landing page for audit data backup and restoration. Change tracker records value changes of monitored properties.

Select **Configure > Financial Services >Change Tracker Management**.

Change Tracker Management

[Archive Change Tracker Table](#)

Warning !!

Archive function will insert records into Change Tracker Archive and delete them from Change Tracker.
Restore function will insert records back into Change Tracker and delete them from Change Tracker Archive.

Please enter past date to Archive or Restore the records.

Archive records from Change Tracker older than

Restore records from archive older than

Snapshot Management: utility for data backup and restoration.

Select **Configure > Financial Services > Sample Data > Snapshot Management.**

FSF Sample DataTools

[Snapshot Management](#)

Create Snapshot

Please provide a name for the Snapshot

Snapshot Name

Manage Existing Snapshots

Select a Snapshot name

NOTE: THIS OPERATION MAY TAKE A LITTLE WHILE. PLEASE BE PATIENT UNTIL THE RESTORE BUTTON IS AVAILABLE.

Snapshot Name

Creating a snapshot:

In the **Create Snapshot** section, enter a name for a new snapshot and then click **Create** to create a new snapshot of data. With this snapshot, you can restore the data state back to this state.

Restoring a snapshot:

In the **Manage Existing Snapshots** section of the **Snapshot Management** tab, select a snapshot name from the drop-down list and then click **Restore**. All sample data is erased, and the sample database is restored to the way it was at the time of the selected snapshot.

Date Shift: Moving dates of accounts and transactions for demo scenarios.

Select **Configure > Financial Services > Sample Data > Date Shift**.

The Date Shift tab allows you to select the number of days you want transaction to shift. In the **Days to shift** box, enter a number, and then click **Shift Dates** to shift all dates that can be shifted.

You can enter a positive number of days to shift dates forward or a negative number of days to shift dates backward.

FSF Sample Data Tools

Date Shift Cascading Date Shift

Please enter the number of days to shift sample data dates. Positive numbers shift to the future, negative numbers shift to the past.

Days to shift: *

Shift Dates

The Cascading Date Shift moves the transactions and account dates relative to today's date. Click Set Date will grab today's date and moves the dates in the transaction and account dates relative to today's date for Open accounts. However, open and closed account dates are not changed.

Externalization of third party data

- Solution overview
- Implementation

Solution overview

The solution is aimed at moving towards a light version of the master profile externalizing third party data. Additionally, the master profile access pattern will be rewired to retrieve third party data only when required. Saving of the master profile will also be updated to save third party data to a different storage.

Master Profile Lite

Detailed external provider data would be removed from the master profile and stored as separate data object there by ensuring master profile is lite.

Clean Loading

Routines to retrieve master profile GetMasterProfile and D_GetMasterProfile will be updated to retrieve external provider data from its new storage on demand.

Synchronization

When a master profile is being saved via SaveWorkFolder API, the external provider data will be written to its own storage and will be cleaned up on master profile. The master profile will house the key properties derived using the external provider information but not the actual data.

Implementation

- Overview
- Externalization on save
- Externalization on access
- On-demand loading
- Synchronization tuning
- Cleanup of data from WorkFolder.pyWorkParty(Customer)

- Externalization of third party provider

Overview

In this first iteration of making the master profile lite all the third provider data, namely eScreening, eAdverseMedia, eCredit, eOFAC, eFraud, eID, eAuthentication and eEnrichment need to be externalized.

This will involve the following main updates:

Externalization on save

The third-party data of a customer will be saved to new database table created for external provider storage using SaveWorkFolder API. Whenever master profile is saved, if there are external provider data recently added for the customer, it will be synchronized to the new table.

Externalization on Access

To ensure that the Master profile is always accessed in the lite mode even in the upgrade scenarios, the master profile access mechanism will be updated to perform the externalization using the existing upgrade module.

On-demand Loading

Master profile access routines will be updated to take a parameter which when true would retrieve the third-party data from its own database table and populate it onto the master profile. These areas primarily being the viewing, reporting and due diligence and customer investigation cases.

Synchronization

Some third-party integrations like the eScreening and eAdverseMedia do not use the standard SaveWorkFolder API for synchronization of corresponding results to the master profile. This issue will be addressed by updating the synchronization engine to

perform selective synchronization of the third-party data and the case types will be updated to populate the summarized data on the ExternalProviderDataPage.

Externalization on save

Currently third-party data of a customer resides in ExternalProviderDataPage of a customer master profile. As part of SaveWorkFolder API, the external provider data needs to be synchronized to the new storage table and cleaned up on the CMP. To facilitate this, SynchronizeExternalProviderData data transform can be updated to invoke the new module described in 2.3 which performs the synchronization and performs clean up on the CMP.

Externalization on access

The externalization on save will ensure that the third-party data is externalized when the master profile is saved, which will work perfectly fine for the new master profiles. Existing master profiles though will have the embedded third-party data which will make it to the case when the data is copied down. Hence, in addition the externalization on save it must also be ensured that the data is externalized, and the master profile cleaned up on load of the master profile.

As mentioned earlier this issue will surface only for the master profiles that are created before the 8.9. Hence the upgrade module can be used for this exercise.

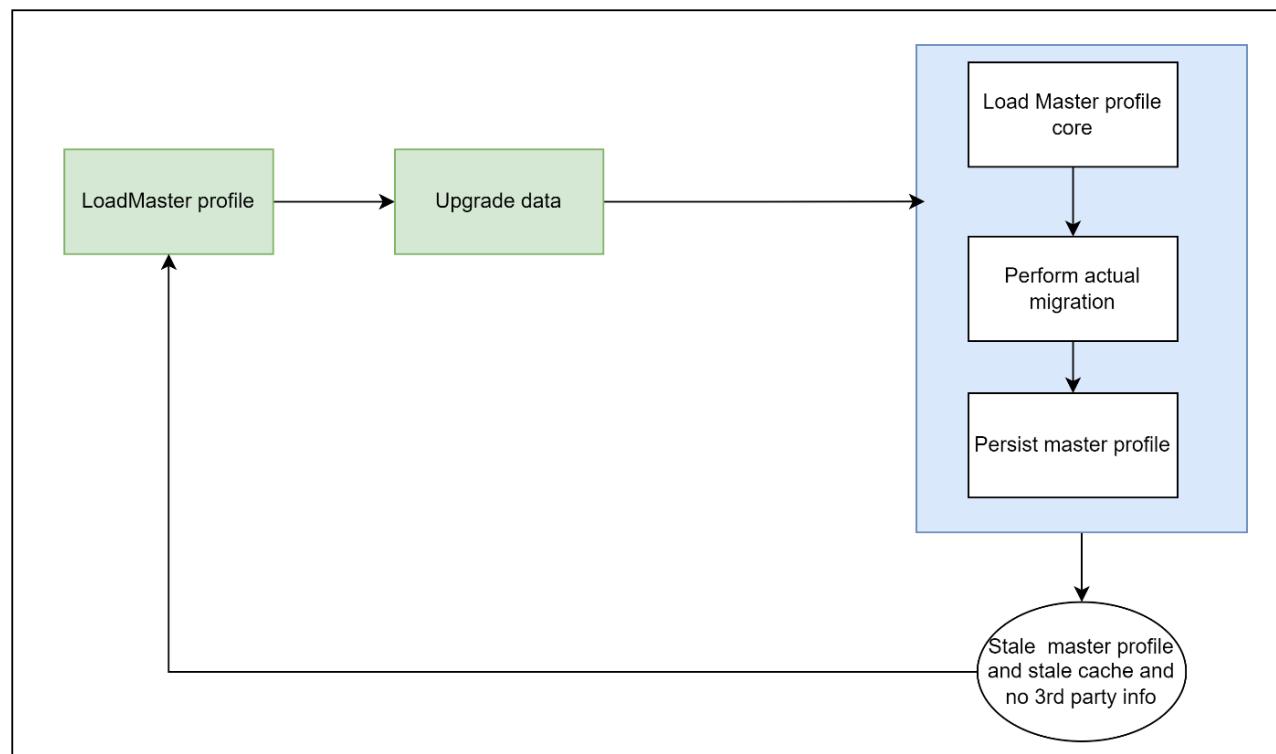
The module is already being invoked on load of the master profile and should be updated to execute an 8.9 upgrade routine. UpgradeDataTo89 routine needs to be extended in PegaFS-Data-Party-MasterProfile class to perform the externalization. As a part of this, externalization needs to be performed in a child requestor (created when load-datapage method is executed) so that the changes can be committed, and the transaction remains separate thereby not affecting the main transaction.

In upgrade routine an activity needs to be invoked to perform the externalization. From the activity, Load-DataPage needs to be invoked, whose source would be another activity which first retrieves the master profile using LoadMasterProfileCore and then

invokes to the main externalization module built in 2.3. That will take care of the externalization and clean-up of the master profile. Then the master profile needs to be persisted and the transaction needs to be committed. Use of PoolID in Load-DataPage needs to be explored.

Post successful migration from the upgrade module, LoadMasterProfile routine needs to be re invoked so that the latest master profile is retrieved.

UpgradeDataTo89Required when rule which determines if the upgrade data transform for 8.9 needs to be invoked, needs an update to consider scenarios where externalization switch was turned on after a master profile was opened post upgrade. To facilitate this, a property needs to be created which would hold whether the third-party data has been migrated to the new table when the switch is turned on. After successful migration the property would be set to true indicating migration is complete and the migration routine is not required to be invoked next time master profile is loaded.



As the third-party data in the master profile is already cleaned up, an explicit cleanup of the data through the ClearScreeningMatches will no longer be required in the following routines and must be eliminated.

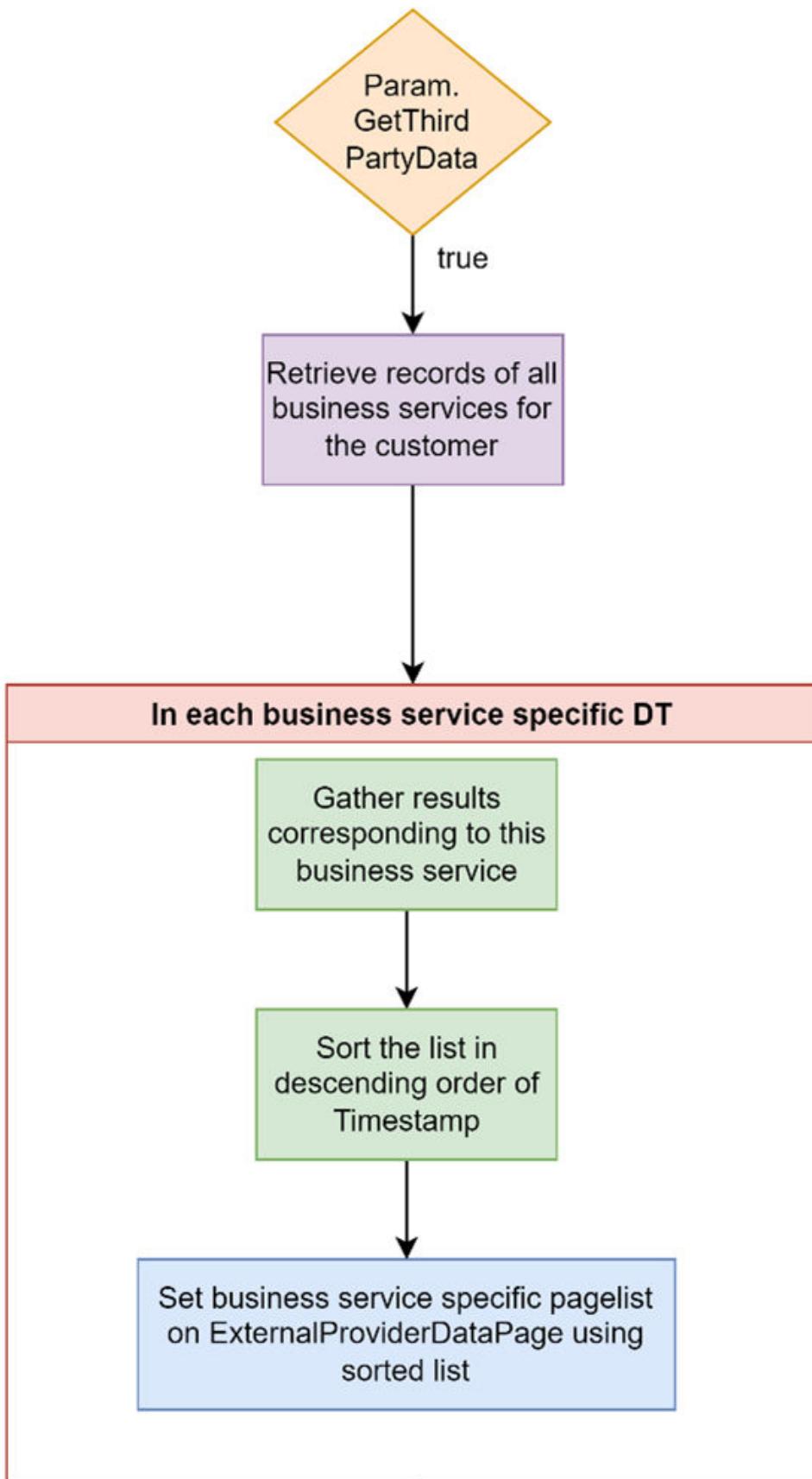
- PegaFS-Work CopyFromMasterProfileToCustomer
- PegaFS-Data-PartyPartyXRef CopyFromRelPartyMasterProfileToRelParty
- PegaCLMFS-Work CopyRPDataFromMaster
- PegaFS-Data-Party EnrichRelatedPartyDataFromMasterProfile
- PegaCLMFS-Work-GlobalKYC PropagateDataToRelatedPartyKYC

On-demand loading

There are few areas in the application that require the full-fledged external provider data, Few of them are Screening case, adverse media case, due diligence cases Customer360PDFSummary and View Third Party Data action from the Customer360. The continuity of these must be catered, to ensure that the data is rightly populated and displayed and used as required.

To achieve this a new routine needs to be built which checks if GetThirdPartyData parameter is set to true, only when it is true this will execute its steps otherwise exits the Data transform. As part of the routine, for each of the business services separate data transforms are to be created and an extension point is to be invoked towards the end.

A data page which retrieves all the business services corresponding to a customer needs to be built and all the routines will use this data page to retrieve business service specific results. In a business service specific routine, it iterates through all the results of the data page and results corresponding to the current business service would be copied to a temporary list. This temporary list is then sorted based on the timestamp with latest entries on top. Once the list is sorted, it would be copied to the actual property on WorkFolder.



A new parameter GetThirdPartyData needs to be added to D_GetMasterProfile and GetMasterProfile routines and in LoadMasterProfile DT, before upgradeData this new routine needs to be invoked to retrieve third party data when necessary.

References of third-party data in the application need to be analyzed and wherever necessary the data needs to be retrieved by passing GetThirdPartyData parameter as true to GetMasterProfile routine. Below is the list of rules which have references to ExternalProviderDataPage, these rules are to be analyzed and wherever required the third-party data needs to be retrieved.

As per preliminary analysis, below are few of the areas where the information is required and corresponding few suggestions are included on how to use the third-party data in these situations, a thorough analysis needs to be performed and solutions are to be derived on how to use the third-party data.

1. GKYC case – Third party data is being used in operationalization, so it better to retrieve the third-party data in the CMP, during GKYC case creation.
2. eScreening and Adverse information cases – Third-party data is required for reusing it, so it better to retrieve the data into CMP during case creation.
3. Onboarding cases – Data is required to determine if customer investigation stage needs to be executed or not and to determine if the eScreening and eAdverse media cases are to be created. For child cases which are getting created, this information might not be required. Currently as pyWorkParty(Customer) is directly being copied to most of the child cases of onboarding cases via propagate DTs, a thorough analysis needs to be performed to check if the child cases needs the third-party otherwise it can be cleared.

Similarly for rest of the scenarios, analysis needs to be performed and based on the scenario appropriate actions need to be taken.

Synchronization tuning

Pega Foundation for Financial Services hosts an extremely well-orchestrated and state of the art API to selectively synchronize the case data to the master profile.

This API is designed with various data categories such as

- GENERAL CLIENT DATA (GCD)
- RELEVANT RELATED PARTY (RRP)
- RISK ASSESSMENT + AML CDD PROFILE (RAP)
- REQUIREMENTS DATA (REQ)

Each of these categories are further divided into subcategories.

For example, for GENERAL CLIENT DATA (GCD) we have the following subcategories:

- GENERAL CLIENT DATA (G)
- CONTACT DATA (C)
- REGULATORY DATA (R)
- PRODUCT DATA (P)
- BUSINESS CASE DATA (B)
- FINANCIAL DATA (F)
- CREDIT DATA (D)
- LEGAL DATA (L)

Currently the external data is synchronized to the master profile through the GCD.G subcategory. Thus, forcing the system to synchronize all the general client data even, this becomes an issue in cases like the eScreening and eAdverseMedia. These cases are designed to retrieve, present, and act on data from the external data providers and hence, cannot guarantee the validity of rest of the general client data.

Hence, the synchronization API cannot be used in such cases which forces them to use the customized mechanisms for external data synchronization.

Secondly, the external data is synchronized to the master profile in a headless manner. That means, every time the synchronization engine is called all the external provider data in the master profile is overridden by the data in the case. Bringing in the possibility of data loss owing to stale data on the case.

In this part of the epic we will be focused at:

1. Replacing the customized synchronization mechanisms with the standard API
2. Eliminating the possibility of data loss

To solve these issues a new subcategory EXTERNAL DATA (E) must be added to the GENERAL CLIENT DATA (GCD) category. The routine `SynchronizeExternalProviderData` must then be moved to `SynchronizeClientData` from `SynchronizeClientDataGeneral` and configured to execute only when the subcategory passed is E.

`SynchronizeExternalProviderData` must then be updated to invoke the module built in 2.3.

Once the routines are ready, the eScreening and eAdverseMedia case types, namely `SynchronizeScreeningData` and `SynchronizeAdverseInformationData` must be updated to invoke the `SaveWorkFolder` by passing the correct category and subcategory. Thus, eliminating the need of custom logic.

To ensure backward compatibility, a DSS needs to be created for customers to choose the subcategory for synchronization of external provider data page. This is to facilitate seamless upgrade scenarios where customers might have synchronized external provider data using GCD.G subcategory, with the new change they will have to update the implementation to use GCD.E subcategory. For new customers, this DSS needs to be set to E, whereas for update customers this will be G. A *when* rule needs to be

created to check for the sub-category before invoking the routines in the synchronization.

Cleanup of data from WorkFolder.pyWorkParty(Customer)

In screening cases, external provider data is not only populated to WorkFolder.ExternalProviderDataPage but is also available under WorkFolder.pyWorkParty(Customer).ExternalProviderDataPage.

After the successful synchronization of the data to the new table, WorkFolder.pyWorkParty(Customer).ExternalProviderDataPage must be cleaned up to remove business service pages. While populating the information using **GetMasterProfile** routines, based on the outcome of the analysis performed in 3.4, if necessary WorkFolder.pyWorkParty(Customer).ExternalProviderDataPage should also be populated to ensure backward compatibility.

Externalization of third party provider

Rule Type	Name	Class
Data Transform	AssesseIDAuthentication	PegaFS-Work
Data Transform	AssesseScreeningCheck	PegaFS-Data-ExtProvid
Data Transform	BusinessServiceAdverseMediaCreateCase	PegaFS-Work-Screening AdverseInformation
Data Transform	BusinessServiceCreditCheck	PegaFS-Work
Data Transform	BusinessServiceeAuthentication	PegaFS-Work
Data Transform	BusinessServiceeIDVerify	PegaFS-Work
Data Transform	BusinessServiceeScreeningCase	PegaFS-Work-Screening
Data Transform	BusinessServiceeScreeningGetResults	PegaFS-Work-Screening
Data Transform	BusinessServiceFraudCheck	PegaFS-Work
Data Transform	BusinessServiceOFACCheck	PegaFS-Work
Data Transform	CalleAuthenticationBusinessService	PegaFS-Work

Rule Type	Name	Class
Data Transform	CalleCreditBusinessService	PegaFS-Work
Data Transform	CalleFraudBusinessService	PegaFS-Work
Data Transform	CalleIDVerifyBusinessService	PegaFS-Work
Data Transform	CalleOFACBusinessService	PegaFS-Work
Data Transform	ClearScreeningMatches	PegaFS-Data-Party
Data Transform	ClearScreeningMatches	PegaFS-Work
Data Transform	ClearScreeningResults	PegaCLMFS-Work-CLM
Data Transform	CopyeIDVerificationInfo	PegaFS-Work
Data Transform	CopyeScreeningRiskToParent	PegaFS-Work
Data Transform	CopyFromMasterProfileToCustomer	PegaFS-Work
Data Transform	CopyFromRelPartyMasterProfileToRelParty	PegaFS-Data-PartyPartyXRef
Data Transform	CopyRPDataFromMaster	PegaCLMFS-Work
Data Transform	CreateOFACScreeningMatch	PegaFS-Data-ExtProvidBusService-ScreeningMatch
Data Transform	EnrichRelatedPartyDataFromMasterProfile	PegaFS-Data-Party
Data Transform	FindAdverseInformationID	PegaFS-Data-Party-MasterProfile
Data Transform	FindCaseSystemId	PegaFS-Data-Party-MasterProfile
Data Transform	GeteIDVerificationInfo	PegaFS-Work
Data Transform	GeteScreeningAssessment	PegaFS-Data-Party
Data Transform	GetOFACScreeningMatches	PegaFS-Work
Data Transform	GetScreeningResultsFromMP	PegaFS-Work

Rule Type	Name	Class
Data Transform	GetWCScreeningMatches	Work-
Data Transform	Init_Global_AML_Entity_Inbound_Mapping	PegaKYC-Data-Type-Policies-Global
Data Transform	Init_Global_AML_Individual_CDD	PegaKYC-Data-Type-Policies-Global
Data Transform	Init_Global_AML_Individual_CDD_Inbound_Mapping_RegData	PegaKYC-Data-Type-Policies-Global
Data Transform	Init_US_AML_Individual_CDD	PegaKYC-Data-Type-Policies-Global-USA
Data Transform	InitializeDataFromThirdParty	PegaCLMFS-Work
Data Transform	InvokeGeneralBusinessServices	PegaFS-Work
Data Transform	MapCustomerDetailsFromThirdParty	PegaFS-Data-ExtProvid-BusService-Client
Data Transform	MapScreeningResultsIntoPEPAndAdverseMedia	PegaFS-Work
Data Transform	MapScreeningResultsIntoSanctionsList	PegaFS-Work
Data Transform	OpenMasterProfile	Work-
Data Transform	OpenMasterProfile	PegaFS-Work-Screening
Data Transform	PreDetermineValidKYCTypes_Global	PegaCLMFS-Work
Data Transform	PreThirdPartyData	PegaFS-Data-Party
Data Transform	PreThirdPartyData	PegaFS-Work
Data Transform	PropagateDataToParent	PegaCLMFS-Work-GlobalKYC
Data Transform	PropagateDataToRelatedPartyKYC	PegaCLMFS-Work-GlobalKYC
Data Transform	PropagateSourceDataToPrimary	PegaCLMFS-Work
Data Transform	ResolveReviewMatch	Data-

Rule Type	Name	Class
Data Transform	ResolveReviewMatches	Work-
Data Transform	ResolveReviewMatches	PegaFS-Work
Data Transform	RetrieveNextScreeningDate	PegaFS-Data-Party
Data Transform	SetupData_EnableSE_G_ECDD_RsltOfNgtvInfoScrng	PegaKYC-Data-Type-Policies-Global
Data Transform	SetupData_EnableSE_G_ICDD_InfoScreeningResult	PegaKYC-Data-Type-Policies-Global
Data Transform	SetupRiskSynchronization	PegaFS-Data-Party
Data Transform	SynchronizeAdverseInformationData	PegaFS-Work-ScreeningAdverseInformation
Data Transform	SynchronizeAdverseInformationDataToParent	PegaFS-Work-ScreeningAdverseInformation
Data Transform	SynchronizeClientDataGeneral	PegaFS-Work
Data Transform	SynchronizeExternalProviderData	PegaFS-Data-Party
Data Transform	SynchronizeExternalProviderData_Ext	PegaFS-Data-Party
Data Transform	SynchronizeExternalRiskeAdverseMedia	PegaFS-Data-Party
Data Transform	SynchronizeExternalRiskeCredit	PegaFS-Data-Party
Data Transform	SynchronizeExternalRiskeFraud	PegaFS-Data-Party
Data Transform	SynchronizeExternalRiskeID	PegaFS-Data-Party
Data Transform	SynchronizeExternalRiskeOFAC	PegaFS-Data-Party
Data Transform	SynchronizeExternalRiskeScreening	PegaFS-Data-Party
Data Transform	SynchronizeRelatedPartyBasicData	PegaFS-Data-Party
Data Transform	SynchronizeScreeningData	PegaFS-Work
Data Transform	SynchronizeScreeningData_Extension	PegaCLMFS-Work-Screening

Rule Type	Name	Class
Data Transform	SynchronizeSpecialMasterProfileData	PegaFS-Work
Data Transform	TestBusinessServices	PegaFS-Data-Party
Data Transform	TriggerAdverseMediaInformationMatches	PegaFS-Work-Screening AdverseInformation
Data Transform	TriggereScreeningGetResults	PegaFS-Work-Screening
Data Transform	TriggerScreening	PegaFS-Work-Screening
Declare Expression	.eCredit	PegaFS-Data-RiskProfile
Declare Expression	.eFraud	PegaFS-Data-RiskProfile
Declare Expression	.eID	PegaFS-Data-RiskProfile
Declare Expression	.eOFAC	PegaFS-Data-RiskProfile
Declare Expression	.eScreening	PegaFS-Data-RiskProfile
Flow Action	ViewECreditData	PegaFS-Work
Flow Action	ViewEFraudCheckData	PegaFS-Work
Flow Action	ViewEIDVerificationData	PegaFS-Work
Flow Action	ViewEOFACData	PegaFS-Work
Flow Action	ViewThirdPartyData	PegaFS-Data-Party
Flow Action	ViewThirdPartyData	PegaFS-Work
Navigation	DisplayIncomeExpenses	PegaFS-Data-Party
Navigation	EnrichCustInfoMenu	PegaFS-Data-Party
Navigation	ViewThirdPartyData	PegaFS-Data-Party
Property	ExternalProviderDataPage	PegaFS-Data-Party

Rule Type	Name	Class
Section	CustomerSummaryExternalAssessmentPDFForInd	PegaFS-Work-CustomerSearch
Section	CustomerSummaryPDFForInd	PegaFS-Work-CustomerSearch
Section	CustomerSummaryPDFForOrg	PegaFS-Work-CustomerSearch
Section	DisplayeAuthenticationDetails	PegaFS-Data-ExtProvid
Section	DisplayECredit	PegaFS-Data-RiskProfil
Section	EnrichSuitability	PegaFS-Data-Party-Ind
Section	EnrichSuitability	PegaFS-Data-Party
Section	ExternalDataInformation	PegaFS-Data-RiskProfil
When	AssessmentExist_eAdverseMedia	PegaFS-Data-Party
When	AssessmentExist_eCredit	PegaFS-Data-Party
When	AssessmentExist_eFraud	PegaFS-Data-Party
When	AssessmentExist_eID	PegaFS-Data-Party
When	AssessmentExist_eOFAC	PegaFS-Data-Party
When	AssessmentExist_eScreening	PegaFS-Data-Party
When	DisplayeAuthentication	PegaFS-Data-Party
When	DisplayeCreditProfile	PegaFS-Data-Party
When	DisplayeEnrichment	PegaFS-Data-Party
When	DisplayeFraudCheck	PegaFS-Data-Party
When	DisplayeIDVerification	PegaFS-Data-Party
When	DisplayeScreening	PegaFS-Data-Party
When	eAdverseInformationPagesExists	PegaFS-Data-RiskProfil
When	eCreditNeedsToBeDone	PegaFS-Work

Rule Type	Name	Class
When	eFraudNeedsToDo	PegaFS-Work
When	eIDVerifyNeedsToDo	PegaFS-Work
When	EnableSE_G_ECDD_RsltOfAdvInfoScrng	PegaKYC-Data-Type-Policies-Global
When	EnableSE_G_ECDD_RsltOfNgtvInfoScrng	PegaKYC-Data-Type-Policies-Global
When	EnableSE_G_ECDD_RsltOfSncntnsScrng	PegaKYC-Data-Type-Policies-Global
When	EnableSE_G_ICDD_AdvInfoScreeningResult	PegaKYC-Data-Type-Policies-Global
When	EnableSE_G_ICDD_IdentifiedPEP	PegaKYC-Data-Type-Policies-Global
When	EnableSE_G_ICDD_SanctionsScreeningResult	PegaKYC-Data-Type-Policies-Global
When	eOFACNeedsToDo	PegaFS-Work
When	HasAdvInfScreeningDateExpired	PegaFS-Data-Party
When	HasAdvInfScreeningMatches	PegaFS-Data-Party
When	HasAdvInfScreeningPages	PegaFS-Data-Party
When	HasPositiveScreeningMatches	Work-
When	HasPositiveScreeningMatches	PegaFS-Work
When	HasScreeningDateExpired	PegaFS-Work
When	HasScreeningMatches	PegaFS-Work
When	HasScreeningPages	PegaFS-Data-Party
When	HasScreeningPages	PegaFS-Work
When	HasScreeningPositiveMatch	PegaFS-Data-Party
When	IsIdVerificationFailed	PegaCLMFS-Work

Rule Type	Name	Class
When	iseKYCVerificationPassed	PegaCLMFS-Work-CLM
When	IsExternalDataRiskExist	PegaFS-Data-RiskProfile
When	IsExternalProviderDataPageExists	PegaFS-Data-Party
When	IsThirdPartyDataCompleted	PegaCLMFS-Work
When	IsThirdPartyDataCompleted	PegaFS-Work
When	NeedToAddOFACRegistryToScreenigMatches	Work-
When	ViewThirdPartyData	PegaFS-Data-Party

Adding a new data source for entity enrichment

Enriching the customer and company data from financial information providers provides many benefits to financial services organizations, including operational efficiencies, more efficient management of credit decisions, verification of identity, increased fraud awareness, provision of more targeted and personalized cross-selling and upselling strategies, and optimization of collection treatments.

Pega Foundation for Financial Services (PFFS) provides a single, unified approach to creating and extending integrations to a list of financial information providers that all Financial Services (FS) applications can access.

There are two parts to enabling a new enrichment data provider in Pega Foundation for Financial Services:

- Adding a new component containing the integration with the data provider (Creating a new component is not covered in this article).
- Implementation in the foundation (application) layer of, for example, construction of input calls to the provider, error handling, and mapping of results.

- **Adding a new component**

- **Implementing the component**

- End results and Use case examples
- References

Adding a new component

Component can be downloaded from Pega Marketplace or enabled in the application rule if already bundled with Pega Foundation for Financial Services. A list of components for every version is available in the Optional Components section of the install guide, available on [Pega Foundation for Financial Services](#) community site. Select your platform version from the Pega Foundation for Financial Services community site.

Before you begin:

1. In the Enabled components section of the application rule, click **Manage components**.

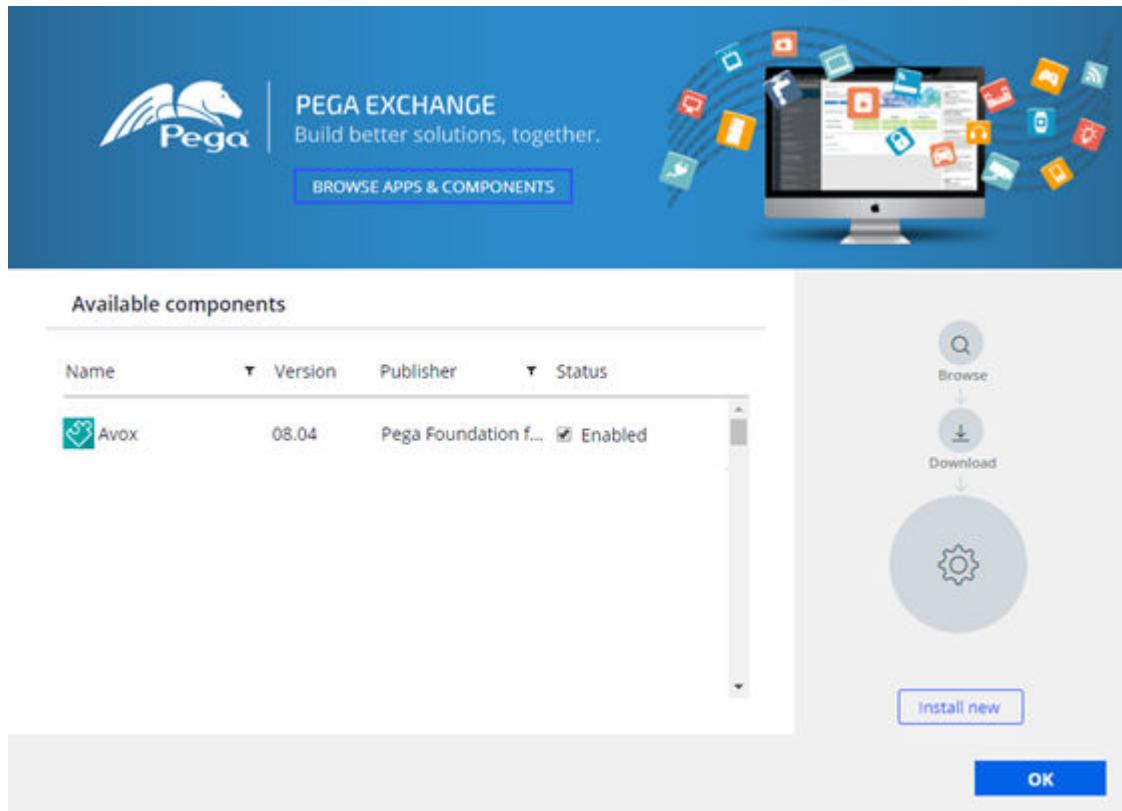
Enabled components	
+ Add component	
Component	Version
No items	

[Manage components](#)

Manage Components

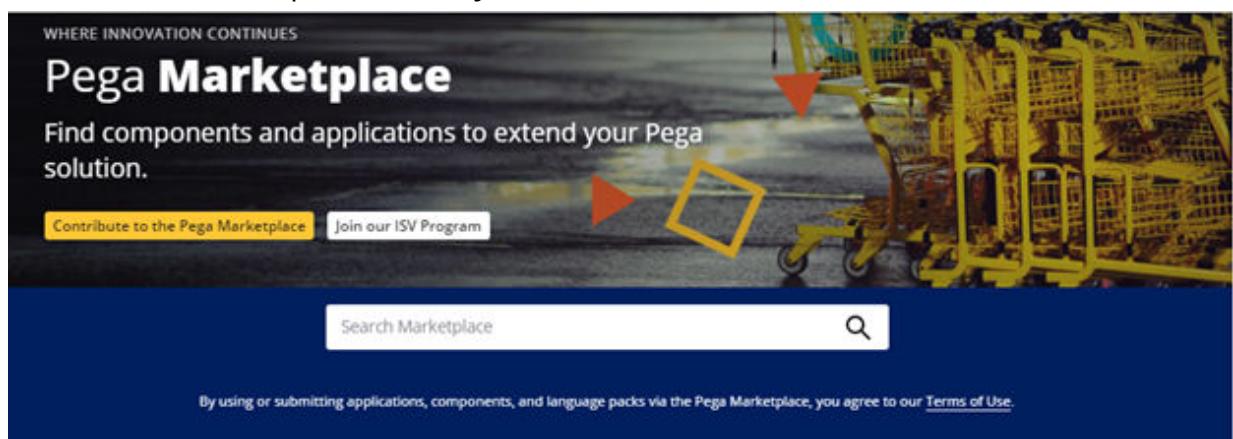
Note: If the desired component is already present in your system, select the **Enabled** check box. Otherwise, check if your component is available and enable or download from Pega MarketPlace by clicking on **BROWSE APPS & COMPONENTS**.

2. Download a component from Pega Marketplace (formerly known as Pega Exchange).



Pega Exchange screen

3. Search for the component that you want to use.



Marketplace

Implementing the component

Verify if PFFS supports your downloaded/enabled component by launching Get eEnrichment Reports case in PFFS. If your component (provider) is listed then 3rd party data provider has been enabled in PFFS and no other setup work needs to be done.

External provider search (EXT-93001)

Get eEnrichment Reports

Select... ▾

Search by name...

Search

Avox

MarkIt

SWIFTKYC

Search for a customer in the provider(s) database.

External provider search

Otherwise, proceed with extending (using save as) rules to add component to entity enrichment processing in PFFS. When adding a new (component) data provider, note the following about Entity Enrichment logic:

- Every component will have 2 functions enabled Search and Details
- All components (data providers) return the same set of data

PFFS class structure of for external data providers has been designed to call each data provider and retrieve results in the unified way. This logic is described below.

There are two data pages in PFFS responsible for searching entities and retrieving entity details. They are described below. In order to add your new data provider, you need to modify data transforms invoked by search entity and entity details data pages.

PegaFS-Data-ExtProvider- is a centralized class for making component calls and mapping returned data back into the Financial Services data structure. There are also specialized child classes that are dedicated to handling enrichment and output mapping.

Class	<i>PegaFS-Data-Ext Provider-BusService-eEnrichment</i>
Description	Class for retrieving search results and entity details for party enrichment.
Data Pages	Unified data pages for searching and for entity details; selects an applicable provider based on input, executes specialized data pages for the provider to perform service calls, and handles output, including errors.

- *D_Trigger_SearchEntity* is a unified data page for searching data providers.
- ClientList page is where results are mapped (see illustration below). This is done in exactly the same for all data providers.

The screenshot shows the Pega Platform Data Page interface for the *D_Trigger_SearchEntity* page. At the top, there are buttons for 'Run', 'Clipboard', 'Trace', and 'Run'. Below that, the 'Run context' section shows 'Thread: Standard'. The 'Parameters' section contains three entries: EntityName (Pega), Provider (AVOX), and MaxResults (5). There is also a checkbox for 'Flush all instances of this data page before execution'. The 'Results' section displays a table of properties and their values. The 'ClientList' row is expanded, showing detailed information for a single record: Avid (36146240), FullName (Pega Ag), LegalEntityIdentifier, Provider (AVOX), TypeOfEntity, pxObjClass (PegaFS-Data-ExtProvider-BusService-Client), and Address.

Property	Value														
pxObjClass	PegaFS-Data-ExtProvider-BusService-eEnrichment														
ptLoadTime	December 23, 2020 10:49:42 AM EST														
ptPageNameBase	D_Trigger_SearchEntity														
ptPageNameHash	_pa7568102242946296pz														
pxDPParameters															
pxSourcePage															
ClientList	<table border="1"> <tr> <td>Avid</td> <td>36146240</td> </tr> <tr> <td>FullName</td> <td>Pega Ag</td> </tr> <tr> <td>LegalEntityIdentifier</td> <td></td> </tr> <tr> <td>Provider</td> <td>AVOX</td> </tr> <tr> <td>TypeOfEntity</td> <td></td> </tr> <tr> <td>pxObjClass</td> <td>PegaFS-Data-ExtProvider-BusService-Client</td> </tr> <tr> <td>Address</td> <td></td> </tr> </table>	Avid	36146240	FullName	Pega Ag	LegalEntityIdentifier		Provider	AVOX	TypeOfEntity		pxObjClass	PegaFS-Data-ExtProvider-BusService-Client	Address	
Avid	36146240														
FullName	Pega Ag														
LegalEntityIdentifier															
Provider	AVOX														
TypeOfEntity															
pxObjClass	PegaFS-Data-ExtProvider-BusService-Client														
Address															

Trigger enrichment search

D_Trigger_SearchEntity calls *Trigger_SearchEntity* data transform where condition to invoke provider specific data page needs to be added.

Note: In the below example, step 1 represents conditions for invoking data page for Avox (D_TriggerAvox_SearchEntity).

D_TriggerAvox_SearchEntity data page calls rules from the component responsible for search.

	Action	Target	Relation	Source
• 1	When	Param.Provider==@getLocalizedText("AVOX","pyLabel").t		
• 1.1	When	IsAvoxComponentPresent		
• 1.1.1	Comment	Invoke Data Page for AVOX		
• 1.1.2	Set	Primary	equal to	D_TriggerAVOX_SearchEntity[EntityName: Par Select values +]
• 2	Otherwise When	Param.Provider==@getLocalizedText("Markit","External")		
• 2.1	When	IsMarkitComponentPresent		
• 2.1.1	Comment	Invoke Data Page for Markit		
• 2.1.2	Set	Primary	equal to	D_TriggerKYCCOM_SearchEntity[EntityName: Select values +]
• 3	Otherwise When	Param.Provider==@getLocalizedText("DB","pyLabel").too		
• 3.1	When	IsDunBradstreetComponentPresent		
• 3.1.1	Comment	Invoke Data Page for DB		

Data Transform - Trigger search entity

- *D_Trigger_EntityDetails* data page for entity details.
- ClientData page is where results are mapped (see illustration below). This is done in exactly the same for all data providers.

Run Data Page: Trigger enrichment subscribed
D_Trigger_SubscribeEntity

Run context
Thread
Standard

Trigger enrichment subscribed was executed using: Standard Thread on RunRecordPrimaryPage page.

Parameter	Value
Provider	Avox
EntityID *	36146240

Flush all instances of this data page before execution

Results

Property	Value
ErrorDescription	No technical infrastructure available for providerAvox
ResultCode	0
pxObjClass	PegaFS-Data-ExtProvider-BusService-eEnrichment
pxLoadTime	December 23, 2020 11:09:42 AM EST
pxPageNameBase	D_Trigger_SubscribeEntity
pxPageNameHash	_pa7569295676720806pz
ClientData	
Avid	36146240
CustomerStatus	Active
EntityClass	
EntityWebsite	Pega Ag
FullName	

Entity Details

D_Trigger_EntityDetails calls **Trigger_EntityDetails** data transform where condition to invoke provider specific data page needs to be added.

Note: In the example below, step 2 represents conditions for invoking data page for Avox (**D_TriggerAvox_EntityDetails**).

D_TriggerAvox_EntityDetails data page calls rules from the component responsible for retrieving entity details.



The screenshot shows the Pega Studio interface for a Data Transform named "Trigger entity details". The rule table has a yellow border around the entire structure. The first row (Comment) and the second row (When) are also highlighted with a yellow border. The rule table columns are: Action, Target, Relation, and Source.

Action	Target	Relation	Source
• 1 Comment	Triggers entity details from external providers		
• 2 When	Param.Provider==@getLocalizedText("AVOX","pyLabel")		
• 2.1 When	IsAvoxComponentPresent		
• 2.1.1 Comment	Invoke Data Page for AVOX		
• 2.1.2 Set	Param.Type	equal to	"AVOX"
• 2.1.3 Set	Primary	equal to	D_TriggerAVOX_EntityDetails[EntityID:Param.]
• 2.1.4 Comment	Set BusServiceIntrospectionPage and source context pointing to AVOX		
• 2.1.5 Set	BusServiceIntrospectionPage.SourceContext	equal to	@Utilities.getLocalizedText("AVOX_Enrichmen")
• 3 Otherwise When	Param.Provider==@getLocalizedText("SWIFTKYC","Ext")		
• 3.1 When	IsSwiftKYCComponentPresent		

TriggerAVOX_EntityDetails

Selection of rule for specific data provider is based on Provider parameter as illustrated above, The end result is invocation of a specific data page > data transform for the provider. Those data page->data transform rules as illustrated with TriggerAvox_EntityDetails example below, handles invocation of data pages from the component and connection errors.

The screenshot shows the Pega Studio interface for a Data Transform named "TriggerAVOX_EntityDetails". The rule table has a yellow border around the entire structure. The first two rows (Set and Comment) are also highlighted with a yellow border. The rule table columns are: Action, Target, Relation, and Source.

Action	Target	Relation	Source
• 1 Set	Param.ConnectorDemo	equal to	@getDataSystemSetting("P")
• 2 Comment	Invoke Data Page for AVOX		
• 3 When	D_Trigger_AvoxDetials[EntityID: Param.]		
• 4 Otherwise When	Param.ConnectorDemo==true		
• 4.1 Apply Data Transform	MapAvoxDummyResponse		
• 5 Otherwise			
• 5.1 Comment	Set error since there is no infrastructure for the provider code		
• 5.2 Set	.ResultCode	equal to	D_Trigger_AvoxDetials[Enti]
• 5.3 Set	.ErrorDescription	equal to	D_Trigger_AvoxDetials[Enti]

Call superclass data transform

End results and Use case examples

Get eEnrichment Reports case in PFFS searches for entities and retrieves entity details based on selected data provider (component). If component has been added/enabled correctly, then it drop menu selection as illustrated below.

External provider search (EXT-14001)

Get eEnrichment Reports

Avox ▾ Pega

Customer name	Address	Legal entity identifier	Entity type	Provider
Pega Ag	Vaduz, LI			AVOX
Pega Capital Corp				AVOX
PEGA-VEL a.s	Krnov, Moravskoslezsko, CZ			AVOX
PEGA Investments Sàrl	Fribourg, Fribourg, CH	506700LK16PC7K3ZBZ27		AVOX
Pega Medical Inc.	Laval, QC, CA			AVOX

Get eEnrichment Reports

Party Details

Get eEnrichment Reports

[< Back to search](#)

PEGA Investments Sàrl Subscribed

AVOX • Last updated 27-Dec-2017

Entity status

Active

Headquarters address

Avenue De Beauregard 12,
Fribourg, CH
Avenue De Beauregard 12,Fribourg, FribourgCH,

Legal form

Regulatory identifier

References

The following are the references:

[Pega Marketplace](#)

[Component submittal form](#)

[Pega Foundation for Financial Services community site](#)

Currency conversion structure

Pega Foundation for Financial Services (PFFS) provides data structure and Create-Read-Update-Delete (CRUD) functions to calculate exchange rates for international credit card transactions. This feature includes ability to store multiple rates per date, retrieve most recent rate or by range for auditing purposes. FS applications use this feature to process international payment transactions.

Table

FSF_REF_CURRENCYEXCHANGERATE (Integration class PegaFS-Int-FSF_REF_CURRENCYEXCHANGERATE).

cer_datetime	cer_from_currency_code	cer_to_currency_code	cer_rate	cer_pxcreatedatetime	cer_pxupdatedatetime	cer_pxupdateoperator	cer_pxcreateoperator
2020-11-23 12:00:00.0	INR	USD	0.013000000	2020-11-23 00:00:00.0	2020-11-23 00:00:00.0	miroj	miroj
2020-11-23 12:00:00.0	JPY	USD	0.009600000	2020-11-23 00:00:00.0	2020-11-23 00:00:00.0	miroj	miroj



Note: Corresponding table FSF_SNAP_CURRENCYEXCHANGERATE contains data backups (snapshots).

Exchange rates table

Data class

PegaFS-Data-CurrencyExchangeRate class contains data structure for currency exchange rate including properties, data transforms for mapping data to and from integration class (table), and data pages for CRUD functions.

PegaFS-Data-CurrencyExchangeRate

- ▼ **Data Model**
 - ▼ **Data Page**
 - ⋮ D_CurrencyExchangeRateOpen
 - ⋮ D_CurrencyExchangeRateSummary
- ▼ **Data Transform**
 - ⋮ GetCurrencyExchangeOpenMap
 - ⋮ OpenMap
 - ⋮ SaveMap
- ▼ **Property**
 - ⋮ CurrencyCodeFrom
 - ⋮ CurrencyCodeTo
 - ⋮ CurrencyExchangeRate
 - ⋮ CurrencyExchangeRateDate
- **SysAdmin**



Note: Information about countries and their currencies is hosted by *PegaFS-Data-Currency <-> PegaFS-Int-FSF_REF_CURRENCY* (table: FSF_REF_CURRENCY).

CurrencyExchangeRate Data class

Data page **D_CurrencyExchangeRateOpen** returns a single record based on primary key. Input parameters are FromCurrencyCode, ToCurrencyCode, DateTime, and Rate. This is for a specific rate information lookup.

Data page **D_CurrencyExchangeRateSummary** returns a list of records based the following input parameters: FromCurrencyCode, ToCurrencyCode, FromDateTime, and ToDateTime. This is to retrieve records for date ranges and currencies for auditing and finding most recent rate.

Use Case

Visa cardholder in the US makes a purchase on an EU based website. The amount paid will be exchanged from USD to Euro by submitting currencies' (To and From) and date/time of transaction to retrieve most recent rate that can be applied to the payment transaction.

Soft delete functionality

Soft-Delete represents functionality to omit data from being displayed or processed without physically deleting records from tables. Table record can be marked as active or inactive, allowing functionality to determine which records to retrieve. Since data is not physically deleted, customer has extra safety net that Soft-Delete can always be reversed by simply changing the status of the record from inactive to active. In Pega Foundation for Financial Services, you can delete the reference data record (Soft-Delete) without physically removing the record from the table.

Tables

Pega Foundation for Financial Services (PFFS) implemented Soft-Delete functionality on several reference data tables:

FSF_REF_COUNTRY (Country codes, names and risk)

FSF_REF_SECTOR (Industry sector codes and descriptions)

FSF_REF_EXCHANGE (International exchange codes and names)

FSF_REF_REGULATOR (International regulator codes and names)

FSF_REF_INDUSTRY (Industry codes and descriptions)

FSF_REF_SUBINDUSTRY (Subindustry codes and descriptions)

FSF_REF_INDUSTRYGROUP (Industry group codes and descriptions)

FSF_RELCODE (Related party codes and descriptions)

PegaFS-Int-FSF_REF_ALTID

PegaFS-Int-FSF_REF_ALTNAME

PegaFS-Int-FSF_REF_BUSINESSCODE

PegaFS-Int-FSF_REF_BUSINESSGOALS

PegaFS-Int-FSF_REF_COUNTERPTYCLASS

PegaFS-Int-FSF_REF_COUNTERPTYTYPE

PegaFS-Int-FSF_REF_CPCLASSTYPEXREF

PegaFS-Int-FSF_REF_CPTYPE_RELTYPEXREF

PegaFS-Int-FSF_REF_CP_BUSCODESXREF

PegaFS-Int-FSF_REF_CP_RELCODEXREF

PegaFS-Int-FSF_REF_CURRENCY

PegaFS-Int-FSF_REF_CUSTPOTENTIALVALUE

PegaFS-Int-FSF_REF_CUSTSEGMENT

PegaFS-Int-FSF_REF_LU_MARKETSEGMENT

PegaFS-Int-FSF_REF_MERCHANTCATEGORYCODES

PegaFS-Int-FSF_REF_PARTYLOCALSUBTYPE

PegaFS-Int-FSF_REF_PARTYSUBTYPE

PegaFS-Int-FSF_REF_PARTYSUBTYPEXREF

PegaFS-Int-FSF_REF_PARTYTYPENAME_MARKETSEGXREF

PegaFS-Int-FSF_REF_PROVINCESTATE

Each table has been extended with **status** column representing if record is **Active** or **Inactive**. Default status value **Active** or **NULL** represents active record.

Possible values for status column are **Active**, **Inactive**, and **NULL**. Values are not case sensitive.

View of FSF_REF_EXCHANGE table with **status** column values set to **Active** for each record.

Each corresponding data class structure has been extended with new status property and mapping of status value to and from tables via CRUD.

e_code	e_name	e_category	e_category_code	e_country	e_status
BAHSE	Bahrain Stock Exchange	Other Exchange	OTHEXCHG	Bahrain	Active
ASE	Australian Stock Exchange	Approved Exchange	APPEXCHG	Australia	Active
BEEURONX	Euronext	Approved Exchange	APPEXCHG	Belgium	Active
BIVA	Bolsa Institucional de ...	Approved Exchange	APPEXCHG	Mexico	Active
BMSE	Bermuda Stock Exchange	Approved Exchange	APPEXCHG	Bermuda	Active
BOLMEXVA	Bolsa Mexicana de Valores	Other Exchange	OTHEXCHG	Mexico	Active
BOSE	Bolivia Stock Exchange	Other Exchange	OTHEXCHG	Bolivia	Active
BRRDJSE	Rio de Janeiro Stock Ex...	Other Exchange	OTHEXCHG	Brazil	Active
BRSPSE	Sao Paulo Stock Exchange	Other Exchange	OTHEXCHG	Brazil	Active

Visually, status column and value can be observed in the Dev Studio Configure > Financial services > Reference Data landing page for country data.

This functionality contains only data structure, logic to handle active vs. inactive records is coming in the future release.

Name	Code (2 digits)	Is OECD?	Is EU?	Is IGA?	IGA Category	Is CRS?	Risk level	Is Sanctioned?	Status	Actions
Afghanistan	AF	Yes	No	Yes	2	No	Specially Sensitive Country	Yes	Active	Actions
Aland Islands	AX	No	No	No		No	Medium Sensitive Country	No	Inactive	Actions
Albania	AL	No	No	No	2	Yes	High Sensitive Country	No	Inactive	View
Algeria	DZ	No	No	Yes	1	No	High Sensitive Country	No	Inactive	Edit
American Samoa	AS	No	No	No		No	High Sensitive Country	No	Inactive	Activate
Andorra	AD	Yes	No	No		Yes	High Sensitive Country	No	Inactive	Actions
Angola	AO	No	No	Yes	1	No	High Sensitive Country	No	Active	Actions
Anguilla	AI	Yes	No	Yes	1	Yes	High Sensitive Country	No	Active	Actions
Antarctica	AQ	No	No	No		No	High Sensitive Country	No	Active	Actions
Antigua and Barbuda	AG	Yes	No	Yes	1	Yes	High Sensitive Country	No	Active	Actions
Argentina	AR	No	No	No		Yes	High Sensitive Country	No	Active	Actions
Armenia	AM	No	No	Yes	2	No	No Sensitive Country	No	Active	Actions

- Class structure
- Soft-delete logic
- Implementation in Financial Services (FS) applications

Class structure

Represents corresponding data and integration classes for each reference data table listed above. Additional information on data integration can be found in PFFS Implementation Guide on [Pega Foundation for Financial Services Community site](#).

Data Class	Integration Class
PegaFS-Data-Country	PegaFS-Int-FSF_REF_COUNTRY
PegaFS-Data-Sector	PegaFS-Int-FSF_REF_SECTOR
PegaFS-Data-Exchange	PegaFS-Int-FSF_REF_EXCHANGE
PegaFS-Data-Regulator	PegaFS-Int-FSF_REF_REGULATOR
PegaFS-Data-Industry	PegaFS-Int-FSF_REF_INDUSTRY
PegaFS-Data-SubIndustry	PegaFS-Int-FSF_REF_SUBINDUSTRY
PegaFS-Data-IndustryGroup	PegaFS-Int-FSF_REF_INDUSTRYGROUP
PegaFS-Data-RelCode	PegaFS-Int-FSF_REF_RELCODE

Soft-delete logic

Data pages returning a list in each corresponding data class have been altered to return only active records or all records based on the two newly introduced parameters.

The parameters are -

1. **ReturnOnlyActive:** If set to true, returns only active records. Records with the status value as 'Active' or 'empty/null' are considered as active records. If set to false/empty, returns all the records.
2. **SectionReadOnly:** This is an internal parameter. This will be effective when ReturnOnlyActive parameter value is true. Value -1 returns all the records, value other than -1 returns only 'Active' records.

Note: Usage of first parameter ReturnOnlyActive is the preferred method to achieve soft delete functionality when there is no intention of making use of platform parameter SectionReadOnly.

Below table illustrates the results for different values of parameter **ReturnOnlyActive** when used alone.

ReturnValueActive param value	Expected results
false	Will return all the records (maintains backward compatibility)
true	Will return the Active records or records with empty/null status

Below is an example data page that shows the addition of two new parameters.

The screenshot shows a 'Data Page: Countries [Available]' with the ID D_CountrySummary and RS: PegaFS:08-08-01. The 'Parameters' tab is selected. The table lists five parameters:

Name	Description	Data type	Required	In/Out	Default value	Smartprompt type	Validate as
1 CountryCode	2 Letter ISO Country Code	String	No	In			
2 CountryName		String	No	In			
3 ReturnNullIfEmpty	ReturnNullIfEmpty	Boolean	No	In	true		
4 ReturnValueActive	If set to true , returns only active rec	Boolean	No	In			
5 SectionReadOnly	This is an internal parameter. This v	String	No	In			

A '+' button is at the bottom left of the table. The table has columns for Name, Description, Data type, Required, In/Out, Default value, Smartprompt type, and Validate as.

Data page

Implementation in Financial Services (FS) applications

In applications where same sections are used both in read only and editable modes, the second parameter **SectionReadOnly** can be made use of. The OOTB platform parameter **SectionReadOnly** needs to be directly passed into this parameter. **ReturnValueActive** needs to be set to true for **SectionReadOnly** parameter to be effective. Below table illustrates the results for various values of the two parameters

ReturnValueActive	SectionReadOnly	Expected results
false	NA	Will return all the records (maintains backward compatibility)
true	-1	Will return all the records and this is the configuration to be used to get all the records in the readonly sections
true	Any value other than -1	Will return only the active records and this is the configuration to be used for editable sections to return only active records

Example configuration for an autocomplete field.

Cell Properties

Requirement: NEVER

List source

Type: Data page

Data page: D_CountrySummary

Search using data page parameter
(Recommended for large data sets)

PARAMETER	VALUE
CountryCode	
CountryName	
ReturnNullIfEmpty	
ReturnActiveOrAll	true
SectionReadOnly	Param.SectionReadOnly

("Param.XXXX" is only available at the section render. Subsequent updates to this page are ignored)

Search results configuration

Cancel **Submit**

Data pages with Soft Delete

Note: The above implementation is purely dependent on the availability of the platform parameter SectionReadOnly and any changes in the platform will impact this. Hence to be used with caution.

Operating structure management (T.O.M)

- What is an Operating Structure?
- Where can the operational structure be found?
- Creating a new operating structure
- Defining taxonomy of the Organization
- Configuring your operating structure

- Defining Sample Organizational Chart

What is an Operating Structure?

- The Operating Structure is an evolution of the standard organization chart, which allows the financial institution to clearly define and differentiate various operational levels that are part of financial institution like business lines, countries, booking entities, and so on.
 - It links the standard organizational structure with the work groups and work baskets. Thus, providing the ability to associate departments and work queues to each operational level thereby making it easy and unambiguous to manage and route work.
 - It controls how and what products the operators can access based on their association to a given operational level.
- Two key facets

Two key facets

Taxonomy:

It is a blueprint that defines strict guidelines and dictates how various operational levels in a financial institution must be associated to each other and how the overall structure of the institution should look like. Users can configure the taxonomy of an organization based on the business needs.

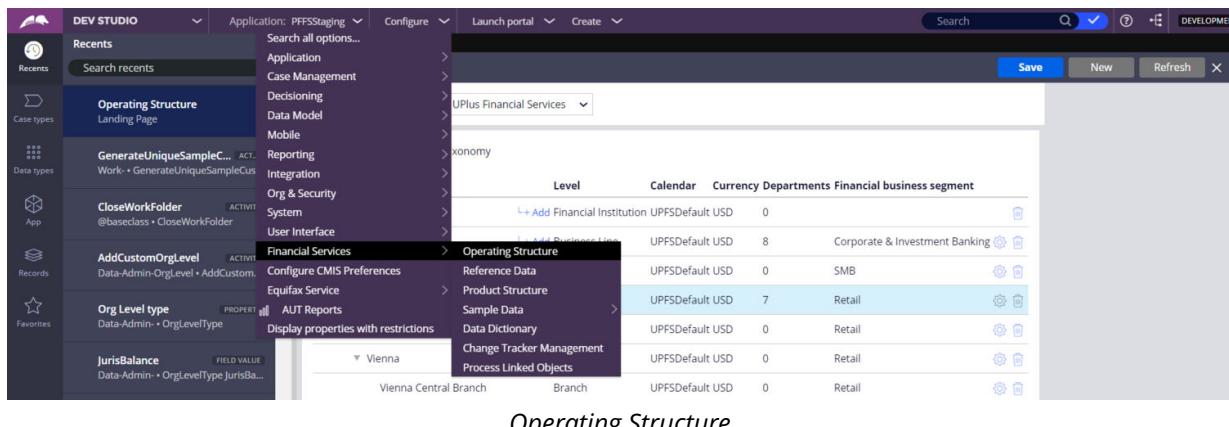
Organizational Chart:

The actual realization of the taxonomy. It is the means to establish the relationships between the different operational levels in a financial institution by realizing the structure defined by its taxonomy.

Where can the operational structure be found?

You can refer UPlus Financial Services taxonomy chart shipped along with Pega Foundation for Financial Services.

1. In Dev Studio click Configure > Financial > Services > Operating > Structure , and select the **Taxonomy** tab.



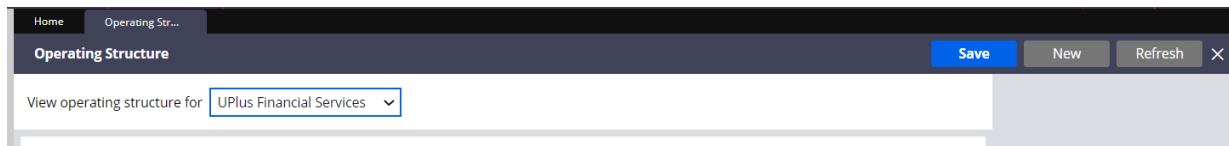
The screenshot shows the Pega Dev Studio interface with the 'Operating Structure' taxonomy table. The table has columns for Level, Calendar, Currency, Departments, and Financial business segment. The data includes:

Level	Calendar	Currency	Departments	Financial business segment
Financial Services	Operating Structure	UPFSDefault USD	0	
Reference Data		UPFSDefault USD	8	Corporate & Investment Banking
Product Structure		UPFSDefault USD	0	SMB
Sample Data		UPFSDefault USD	7	Retail
Data Dictionary		UPFSDefault USD	0	Retail
Change Tracker Management		UPFSDefault USD	0	Retail
Process Linked Objects		UPFSDefault USD	0	Retail

Below the table, there is a section titled 'Vienna' with 'Vienna Central Branch' and 'Branch' listed under it.

Operating Structure

2. In the View Operational Struture for field, select UPlus Financial Services and click Submit. Review the simulated taxonomy as an example of how to set up your taxonomy.



The screenshot shows the 'Operating Structure' view in Pega Dev Studio. The 'View operating structure for' dropdown is set to 'UPlus Financial Services'. The interface includes standard buttons for Save, New, Refresh, and Close.

UPlus Financial Services

Creating a new operating structure

The operating structure is an asset that can be shared across different Financial Services applications. Check first if you have one for your organization already in the system.

If you do not have any, you can create a new one by completing the following steps:

1. In Dev Studio click Configure > Financial Services > Operating Structure.
2. Click **New**.

3. In the **Name** field, enter a name for the bank. The ID field is automatically populated with a 6-character ID.
4. In the **Top-level class** field, enter the class name that will be used to maintain the enterprise class structure.
5. **Optional:** Modify the **Calendar** and **Currency** fields.
6. Click **Submit**.



Note: The system creates a new organization entity Data-Admin-Organization that is ready to sustain a new operating structure.

Defining taxonomy of the Organization

Use the Taxonomy tab to set up the blueprint for your organizational chart. The taxonomy chart defines the types of entities that appear in your organizational chart. All taxonomies should contain at least four default levels (Financial Institution, Business Line, Jurisdiction, and Balance Sheet), but additional levels can be created as required.

The table below shows the different levels that can be used to build your taxonomy.

1. Click Configure > Financial Services > Operating Structure and select your organization.
2. Click the Taxonomy tab.
3. In the new taxonomy, click Specialize and select the type of business line to create.
4. Click Add to assign a new operational level to the business line.
5. Enter a name for the new level and select the level type.
6. Click Save.

Configuring your operating structure

Define the specific business lines, countries, branches, and booking entities that participate in your application, as well as the workgroups and workbaskets used by the application to manage and route the work.

A Sample Operating Structure model is shipped with Pega Foundation for Financial Services to define how to route and organize the work within an organization. The model includes two main components:

Defining Taxonomy of sample Operating Structure

Operational level	Level type	Access groups
Financial Institution	Business	Yes
Business Line	Business	No
Business Line (CIB)	Business	No
Business Line (Retail)	Business	No
Business Line (SMB)	Business	No
Business Line (Wealth)	Business	No

Operating Structure

Financial Institution: Represents organization for which the taxonomy is being configured. This should always be the first level of the Taxonomy.

Business Line: Represents the business segment like CIB, Retail, Wealth, and so on. This should always be the second level in the Taxonomy.

Region: Configurable level generally used to allow the grouping of jurisdictions based on geographies, for example, APC, EMEA, and so on.

Country: Operational level of type Jurisdiction that represents the country where the booking entity is located.

Branch: Operational level that represents the booking entity. This level should always be the last level in the operational structure.

Specialize: This link allows specialization of the business lines for Commercial, Retail, Corporate and Investment banking, SMB, and Private Wealth Management.

Level type: Define the type of the operational level. The following are the type of levels shipped with Pega Foundation for Financial Services:

- Business
- Geographical
- Jurisdictional
- Balance Sheet
- JurisBalance (Jurisdictional + Balance Sheet)

The following table shows the different levels that can be used to build your taxonomy:

Level	Description	Examples
Financial Institution	Node that represents the top-level entity (your organization). Cannot be modified.	UPlus Financial Services
Business Line	Represents the different types of business that your organization manages.	Retail Banking, Commercial Banking
Geographical	Main geographical regions in which your business is present.	Americas, EMEA, APAC

Jurisdictional	Jurisdictions or countries where your business is present and that may have operated and regulatory implications.	Australia, UAE, United States
Balance Sheet	Branches or booking entities used to commercialize your products.	New York, Sydney, Madrid

Access groups: Define the access groups that are associated to an operational level.

Defining Sample Organizational Chart

- **Organizational Chart**
- **Business Validations for selection of level types**
- **Users and Access groups**
- **Utilization in CLM/KYC**
- **Removal of Taxonomy Restrictions**

Organizational Chart

The actual realization of the taxonomy. It is the means to establish the relationships between the different operational levels in a financial institution by realizing the structure defined by its taxonomy.

Global Markets: The primary business line of the organization that follows the taxonomy defined for the default business segment.

Global Investment Banking: The business line of the Organization that manages investment banking and follows the specialized taxonomy defined for the Corporate and Investment banking business segment.

Level: Operational level defined in the taxonomy of the organization.

Calendar: Calendar configured at the operational levels.

Currency: Currency configured at the operational levels.

Departments: Number of departments that are configured for each operational level.

Business segment: The segment specialized taxonomy that the business line and its child nodes follow. Active stands for the base business level. If the business needs additional business segments, these can be added by creating field values for Business Segment field.

- Note:** Finalize the taxonomy of the financial institution before you start building the organization chart. Changes in the taxonomy will require deletion of the organizational chart, this will leave behind a slew of stray rules.

Business Validations for selection of level types

1. Geographical cannot be added above business level types.
2. Bottom most node always represent 'geographical' level type. It cannot be modified.
3. Org structure must have either a single jurisdictional and Balance Sheet level (or) both jurisdictional and balance sheet level independently.

You can configure as stated below.

For example, Business > Jurisdictional > Balance sheet > Geographical

Business > Jurisbalance > > Geographical

4. Operation structure cannot have more than a single balance sheet level/ single jurisdictional level. Below examples are not allowed.

For example, Business > Jurisdictional > Balance sheet > Balance sheet > geographical

For example, Business > Jurisdictional > Jurisdictional > Balance sheet > geographical

Users and Access groups

- Operators must be configured to be associated to a unit and work group.
- Each work group must be manually updated to have a manager.
- Access groups must be configured to have required roles application and so on.

Utilization in CLM/KYC

Routing: The operating structure provides the ability to create work groups and work queues for each operational level. The work objects can be routed to work queues/ work parties for their actions.

Product Access: The operational structure is used to define the products that an operational node can onboard for a customer. It also dictates the jurisdiction and balance sheet levels that the operator can access, thereby defining the products that they can onboard.

The products that the booking entity can deal with is defined by the inclusion and exclusion tables.

- Inclusion table lists the access of an operational level to a product, implicitly providing access to all the balance sheet nodes under that operational node.
- Exclusion table is used to restrict an operational node from accessing a product that it may have gained an implicit access through the inclusion table

- For a more detailed configuration on inclusion or exclusion tables, please refer [Connect FS products to system of record](#).

Removal of Taxonomy Restrictions

In Pega Foundation for Financial Services '23, the operating structure feature is designed to accommodate any intermediary levels in the Taxonomy and thereafter extending the Org chart with Operational levels in sync with the Taxonomy. Removing the taxonomy restrictions would help customers to extend the Org chart as and when it is required.

Any FI can change the taxonomy as and when needed and update the org structure according to the alterations without being affecting the existing Operating Structure and its associated artefacts (WQ, WGs).

Reusability of existing org charts to operating structure

The target operating model (T.O.M) is enhanced to reuse the existing Organizational charts which are created either through New Application Wizard or Organization landing page. The user must convert the organizational chart in order to use it as 'Operating Structure'. The Convert button is available on header of the Operating Structure landing page.

- [Traditional methods of creating an Org chart \(OOTB\)](#)
- [How to reuse the existing Organizational Charts?](#)

Traditional methods of creating an Org chart (OOTB)

The following are the possible ways of creating Org charts in an application during implementation layer:

1. Go to Application > New Application
2. Go to Configure > Org & Security > Organization > Organizational Chart

When the Org charts along with units/ divisions are created by either of the methods listed above, during CLM implementation at client side, all these artefacts can now be reused as-is in Operating Structure. As Operating structure feature is tightly coupled with the CLM onboarding cases, clients can reuse the Org related artefacts without the need of manual deletion and recreating them in Operating Structure Landing page.

How to reuse the existing Organizational Charts?

There are two steps to achieve this objective.

- Convert the Organizational chart to Operating Structure
- Import the existing org units
- **Converting Org chart to Operating structure**
- **Importing Org units or divisions to Operating structure**

Converting Org chart to Operating structure

Follow these steps to convert the orga chart to operating structure.

1. Go to Operating Structure landing page.
2. Select the desired organization from the drop-down "View Operating Structure for."
3. Click on Convert button. You will be prompted to enter the fields such as 'Top level class', 'Calendar', 'Currency' in Convert org chart modal.
4. Click Submit. The Org chart is now converted to Operating Structure and can be reused further to extend it. You may also notice that the Org divisions/units/sub-units which were created for this Org chart would also appear in the Organizational chart tab.
5. Go to Taxonomy tab to define the various business segment according to your institutional requirements.

Importing Org units or divisions to Operating structure

Follow these steps to import the org units/divisions to operating strucutre.

1. Post setting up the taxonomy, go to Organizational chart tab for importing the Org divisions/units/sub-units.
2. Click on Update link to map the org divisions/units/sub-units to financial business segments that were created in the Taxonomy tab.
3. Map the values to desired financial business segment.

Result:

What do you expect the user to see or be able to do after they complete this design pattern?

Configuring data for product availability processing in PFFS

This is a set of instructions to setup product data in Pega Foundation for Financial Services (PFFS) and explain its usage in Product Selection process in CLM/KYC and SAFS applications. Product data and rules that determine product availability are hosted by PFFS.

Determining the availability of Financial Services (FS) products is critical in the following FS applications:

- Sales Automation for Financial Services (SAFS) - the availability of products is used in the creation of Opportunities.
- Client Lifecycle Management for Financial Services (CLM) - the availability of products is determined for the product selection onboarding process.

Selection of Financial Services (FS) products is dependent on the availability of the products, which is based on the location of the organizational unit within the corporate structure of the operator (the user executing the case). The availability of products is based on their inclusion, which is specified as a list of product IDs or entire lines of

business that are marked as "allowed" for selection at the operator's location, as well as their exclusion, which is represented by an exception list of products that are marked as "not allowed" for selection. For example, Global Markets' business line contains 3 products: A, B, and C. Inclusion specifies that all of Global Markets' products are available in a particular corporate structure location (e.g. region X). Exclusion specifies that certain sub-locations within region X (For example, specific countries, booking entities) don't allow product C to be selected. The result is a final list of products that are displayed for selection after the application of inclusion and exclusion conditions.

The screenshot shows a user interface for selecting products. At the top, there is a search bar with the placeholder text "Search products by jurisdiction and booking entity and select one or many products." Below the search bar are two dropdown menus: "Netherlands" and "UPlus-Amsterdam". To the right of these dropdowns is a navigation bar with page numbers from 1 to 10 and a "next" button. The main area displays a table with four columns: "Product type", "Business line", "Product category", and "Select". The table contains five rows of data:

Product type	Business line	Product category	Select
Asset Finance	Global Corporate Finance	Global Corporate Finance	Select
Asset Based Lending	Global Corporate Finance	Global Corporate Finance	Select
Debtor Protection	Global Corporate Finance	Global Corporate Finance	Select
Invoice Finance	Global Corporate Finance	Global Corporate Finance	Select
Mergers and Acquisitions	Global Corporate Finance	Global Corporate Finance	Select

Add products screen

Note: Operators making the product selection have the following organizational unit path:



Edit Operator ID: CLMFSCIBSysAdmin
ID: CLMFSCIBSysAdmin RS: PegaCLMFSCIB [Edit]

Profile Work Security History

Routing

Organizational unit
UPFS / GM / UPFS-GM-EMEA **Update**

Operator selection

- Current state of inclusion and exclusion of products

Current state of inclusion and exclusion of products

- Organizational and product structures
- Product matrix and inclusion or exclusion of cross-reference tables
- Methods of deriving available products
- Class structure

Organizational and product structures

A dependency exists between the availability of a product and the organizational unit (product selection of CLM) of the operator, and/or the product structure (SAFS's products). Both structures are managed by clicking **Configure > Financial Services** and then clicking the Operating Structure or Product Structure tabs.

Entity name	Level	Calendar	Currency	Departments	Financial business segment
UPlus Financial Services	L+ Add Financial Institution	UPFSDefault USD	0		
Asset Management	L+ Add Business Line	UPFSDefault USD	8	Corporate & Investment Banking	 
Global Business Banking	L+ Add Business Line	UPFSDefault USD	0	SMB	 
Global Consumer Banking	L+ Add Business Line	UPFSDefault USD	7	Retail	 
Global Markets	L+ Add Business Line	UPFSDefault USD	9	Corporate & Investment Banking	 
Americas	L+ Add Region	UPFSDefault USD	6	Corporate & Investment Banking	 
APAC	L+ Add Region	UPFSDefault USD	6	Corporate & Investment Banking	 
EMEA	L+ Add Region	UPFSDefault USD	7	Corporate & Investment Banking	 
UAE	L+ Add Country	UPFSDefault USD	6	Corporate & Investment Banking	 
Austria	L+ Add Country	UPFSDefault USD	6	Corporate & Investment Banking	 
Belgium	L+ Add Country	UPFSDefault USD	6	Corporate & Investment Banking	 
UPlus-Brussels	Booking Entity	UPFSDefault USD	1	Corporate & Investment Banking	 
Bulgaria	L+ Add Country	UPFSDefault USD	6	Corporate & Investment Banking	 
Switzerland	L+ Add Country	UPFSDefault USD	6	Corporate & Investment Banking	 

Operating Structure

Follows a defined taxonomy where:

- UPlus Financial Services = Financial Institution
- Global Markets = Business Line
- EMEA = Region
- Belgium = Country
- UPlus-Brussels = Booking Entity

Operating Structure

View operating structure for UPlus Financial Services ▾

Organizational chart [Taxonomy](#)

You are in the CLMFS Application. Under that context All Financial Institutions should contain four default levels: Financial Institution (business-like level), Business Line (business-like level), Country (jurisdictional level) and Booking Entity (Balance Sheet level) as bottom one. The last two are fully editable. You can add new Taxonomy Levels as required by your Financial Institution.

Operational level	Level type	Access groups
Financial Institution	Business	Yes
Business Line	Business	No
Business Line (CIB)	Business	No
Region	Geographical	No
Country	Jurisdictional	No
Booking Entity	BalanceSheet	No
Business Line (Retail)	Business	No
Business Line (SMB)	Business	No
Business Line (Wealth)	Business	No

Operating Structure Taxonomy

View product structure for UPlus FS Product Operations ▾

Product Chart [Taxonomy](#)

Entity name	Level	Calendar	Currency	Departments	Financial business segment
UPlus FS Product Operations	└ Add Financial Institution	USDefault	USD	0	
Product Operations	└ Add Business Line	USDefault	USD	1	
Corporate and Investment Banking	└ Add Segment	USDefault	USD	1	
Global Corporate Finance	└ Add Line of business	USDefault	USD	1	
Global Corporate Finance	└ Add Product Category	USDefault	USD	0	
Asset Based Lending	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Asset Finance	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Debt Protection	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Invoice Finance	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Mergers and Acquisitions	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Project and Export Finance	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Payroll Finance	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Strategic Finance	└ Add Product Type	USDefault	USD	0	Corporate & Investment Banking

Product Structure

Follows a taxonomy structure where:

- Taxonomy can be expanded to fit the specific product structure of the organization.

View product structure for UPlus FS Product Operations			
Product Chart		Taxonomy	
All Financial Institutions should contain three default levels: financial institution, segment and a renameable bottom level.			
Operational level		Level type	Access groups
▼ Financial Institution	↳ Add	Business	Yes
▼ Business Line	↳ Add	Business	No
▼ Segment	↳ Add	Business	No
▼ Line of business	↳ Add	Business	No
▼ Product Category	↳ Add	Business	No
▼ Product Type	↳ Specialize	↳ Add	Business
Branch		Geographical	No

Product Structure Taxonomy

- Operating Structure and Product Structure items created/edited in the UX are saved in the PR_DATA_ADMIN table.
- UPFS is the financial institution abbreviation representing Operating (Organization) Structure.
- UPFSPD is abbreviation represents Product Structure.

The path listed in PYORGUNIT column matches with data in Inclusion and Exclusion cross-reference tables, to determine product availability in product selection:

pyname	pyorgdivision	pyorgunit	pyorganization
BratislavaWealthOffice	PWM	UPFS-PWM-SK-BR-BROff	UPFS
Boston	PWM	UPFS-PWM-US-Boston	UPFS
BeaconHillOffice	PWM	UPFS-PWM-US-Boston-BHOFF	UPFS
NationalOffice	PWM	UPFS-PWM-US-Boston-NatOff	UPFS
NewYorkCity	PWM	UPFS-PWM-US-NYC	UPFS
CentralParkOffice	PWM	UPFS-PWM-US-NYC-CPOff	UPFS
Cards	PDOPB	UPFSPD-PDOPB-RET-CARDS	UPFSPD
AssetBasedLending	PDOP	UPFSPD-PDOP-CIB-GCF-GCF-ABL	UPFSPD
AssetFinance	PDOP	UPFSPD-PDOP-CIB-GCF-GCF-AF	UPFSPD
SydneyOperaOffice	PWM	UPFS-PWM-AU-Sydney-SOOff	UPFS

Data Admin Table

The PR_DATA_ADMIN table also contains operator, workgroup and workbasket records. Workgroups and Workbaskets are created automatically for each new organizational unit created in the UX for the above-defined Operating (Organization) and Product structures:

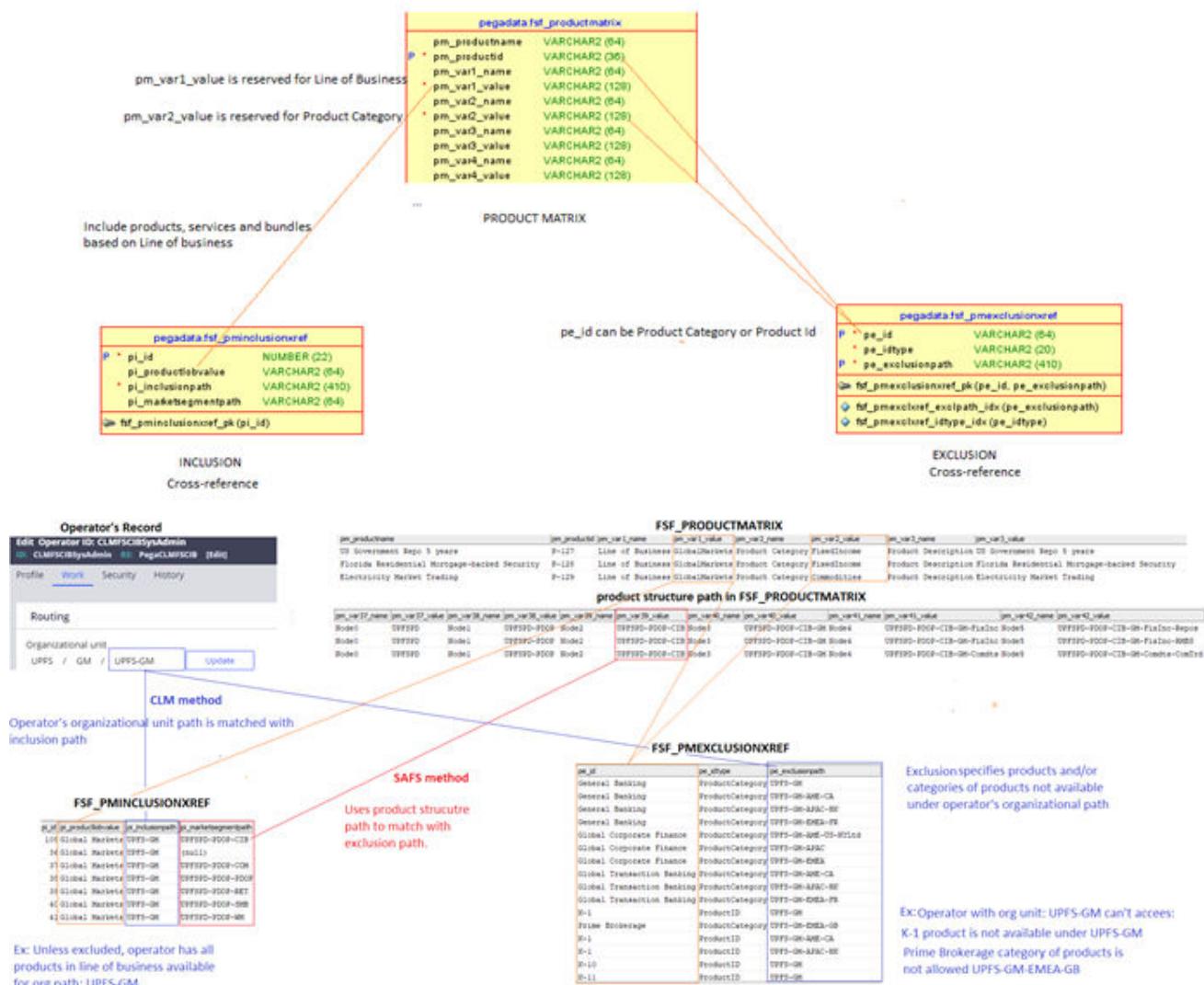
pxinsname	pxobjclass
UPFSPD_PDOP_WM_PRDOPR_DEFAULT	Data-Admin-WorkBasket
UPFSPD_PDOP_WM_WMAPPROVERWB	Data-Admin-WorkBasket
UPFSPD_PDOP_WM_WMAUTHORWB	Data-Admin-WorkBasket
UPFSPD_PDOP_WM_WMPUBLISHWB	Data-Admin-WorkBasket
UPFS_GM_GSCREENING REVIEW	Data-Admin-WorkBasket
UPFS_GM_LEGAL REVIEW	Data-Admin-WorkBasket
UPFSPD_PDOP_COM_DEP_PRDOPR	Data-Admin-WorkGroup
UPFSPD_PDOP_COM_LEND_PRDOPR	Data-Admin-WorkGroup
UPFSPD_PDOP_COM_PRDOPR	Data-Admin-WorkGroup
UPFSPD_PDOP_PRDOPR	Data-Admin-WorkGroup
<i>Admin Operator</i>	

Product matrix and inclusion or exclusion of cross-reference tables

- The FSF_PRODUCTMATRIX table contains basic information records on products, services and product bundles. To generalize, let us refer to all of these as product records. Each product record belongs to a product category and a line of business.
- The FSF_PMINCLUSIONXREF table contains lines of business indicating that all product records for the given line of business are allowed.
- The FSF_MPEXCLUSIONXREF table contains records that block specific product ids or all products belonging to a product category from being allowed.

Basic formula: Retrieve a list of product records from PRODUCTMATRIX for a specified line of business (INCLUSION), and omit from that list all or some products from a specified product category (EXCLUSION).

INCLUSION – EXCLUSION = List of available products from the PRODUCTMATRIX



Data Inclusion and Exclusion

Inclusion-Exclusion

- Data for products, services and bundles contained in FSF_PRODUCTMATRIX table can be inserted or edited using the Product Designer for Financial Services (PDFS) application.
- Data in FSF_PMINCLUSIONXREF and FSF_PMEXCLUSIONXREF is inserted/updated manually using SQL editor.

Methods of deriving available products

CLM Method



Matches the organizational structure path in the operator's record with the inclusion path in the FSF_PMINCLUSIONXREF (INCLUSION) table. By default, all products in a matched line of business are available unless specified in FSF_PMEXCLUSIONXREF (EXCLUSION). The FSF_PMEXCLUSIONXREF table specifies products or product categories (all products within a category) that are not allowed.

INCLUSION – EXCLUSION = AVAILABLE LIST OF PRODUCTS

INCLUSION = AVAILABLE LIST OF PRODUCTS (when products are not excluded)

SAFS Method

Matches the product structure path in FSF_PRODUCTMATRIX with the market segment path in FSF_PMINCLUSIONXREF (INCLUSION) table. All products in a matched line of business are available.

INCLUSION = AVAILABLE LIST OF PRODUCTS

Class structure

Integration class: *PegaFS-Data-ProductMatrixInclusionXref*

Report definition: *GetLOBByLocation* takes the Organizational structure path of the operator and matches it with the inclusion path.

Integration class: *PegaFS-Data-ProductMatrixExclusionXref*

Report definition:

- *GetExclusionsByProductId* takes the Organizational structure path of the operator and searches for *type: product* to return a list of product IDs.
- *GetExclusionsByPMVar2Value* takes the Organizational structure path of the operator and searches for *type: product category* to return a list of product categories.

Integration class: *PegaFS-Int-FSF_PRODUCTMATRIX*

Data class: *PegaFS-Data-ProductMatrix*

Data pages:

- *D_AllowedProductByLocation* retrieves available products. Invokes activity *GetAllowedProductsByLocation*.
- *D_AvailableLOB* retrieves available lines of businesses from inclusion.
- *D_AvailableProductCategory* retrieves list of product categories not listed in exclusion.

Activity: *GetAllowedProductsByLocation* applies formulas based on the following input parameters:

- INCLUSION – EXCLUSION = AVAILABLE LIST OF PRODUCTS
- INCLUSION = AVAILABLE LIST OF PRODUCTS (when products are not excluded).

Creating a sample financial product

This design pattern describes the process and thinking when creating sample products primarily using the Product Designer for Financial Services (PDFS).

The PDFS application is an example of how clients can use Pega's low code case management to define and manage a product's lifecycle. PDFS's sample products have two purposes:

- Populating the PDFS sample application with a set of realistic products so that the dashboards, catalog, and library have decent data to show during a PDFS demo to a customer.
 - Working as a common set of products that can be used by PDFS's consuming applications: PMFS, SAFS, CLM.
-
- **Creating sample products using PDFS**
 - **Sample product structure**
 - **Product catalog ruleset**

- **Attributes**
- **Price**
- **Eligibility**
- **Benefits**
- **Templates**
- **Additional guidelines**
- **Products**
- **Bundles**

Creating sample products using PDFS

Follow these steps when creating sample products:

1. Create a sample product structure.
2. Ensure that the Product Catalog ruleset is set to Demo.
3. Add any additional attributes that may be needed.
4. Create the required price rules.
5. Create the required eligibility rules.
6. Create the required benefit rules.
7. Create the appropriate template(s).
8. Create the products (accounts/services).
9. Create the bundles.

Sample product structure

The sample product structure should be set up for the market segment that you are creating products for. Below is a snapshot of a sample product structure that was set up in a PDFS application:

View product structure for UPlus FS Product Operations.					
Product Chart		Taxonomy			
Entity name	Level	Calendar	Currency	Departments	Financial business segment
UPlus FS Product Operations	Add Financial Institution	USDefault	USD	0	
Product Operations	Add Business Line	USDefault	USD	1	
Corporate and Investment Banking	Add Segment	USDefault	USD	1	
Global Corporate Finance	Add Line of business	USDefault	USD	1	
Global Corporate Finance	Add Product Category	USDefault	USD	0	
Asset Based Lending	Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Asset Finance	Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Debt Protection	Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Invoice Finance	Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Mergers and Acquisitions	Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Project and Export Finance	Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Payroll Finance	Add Product Type	USDefault	USD	0	Corporate & Investment Banking
Strategic Finance	Add Product Type	USDefault	USD	0	Corporate & Investment Banking

Sample Product Structure

The market segments are Retail, Small Medium Business, Commercial, and Corporate & Investment Banking. Ensure that the taxonomy is created and the product structure is populated with appropriate values. Once the product structure has been created, then create the following:

- The appropriate workbaskets
- The Operator IDs, using the following naming standard: Operator id@ 'market segment'.uplusfs.com

The operators that need to be created are:

- Desiree.Dixon@... (Product Designer)
- Mark.Martinez@... (Product Marketer)
- Manny.May@... (Manager/Approver)
- Pam.Perkins@... (Power user – has all roles)
- Dev.Darsha@... (developer/sysadmin)

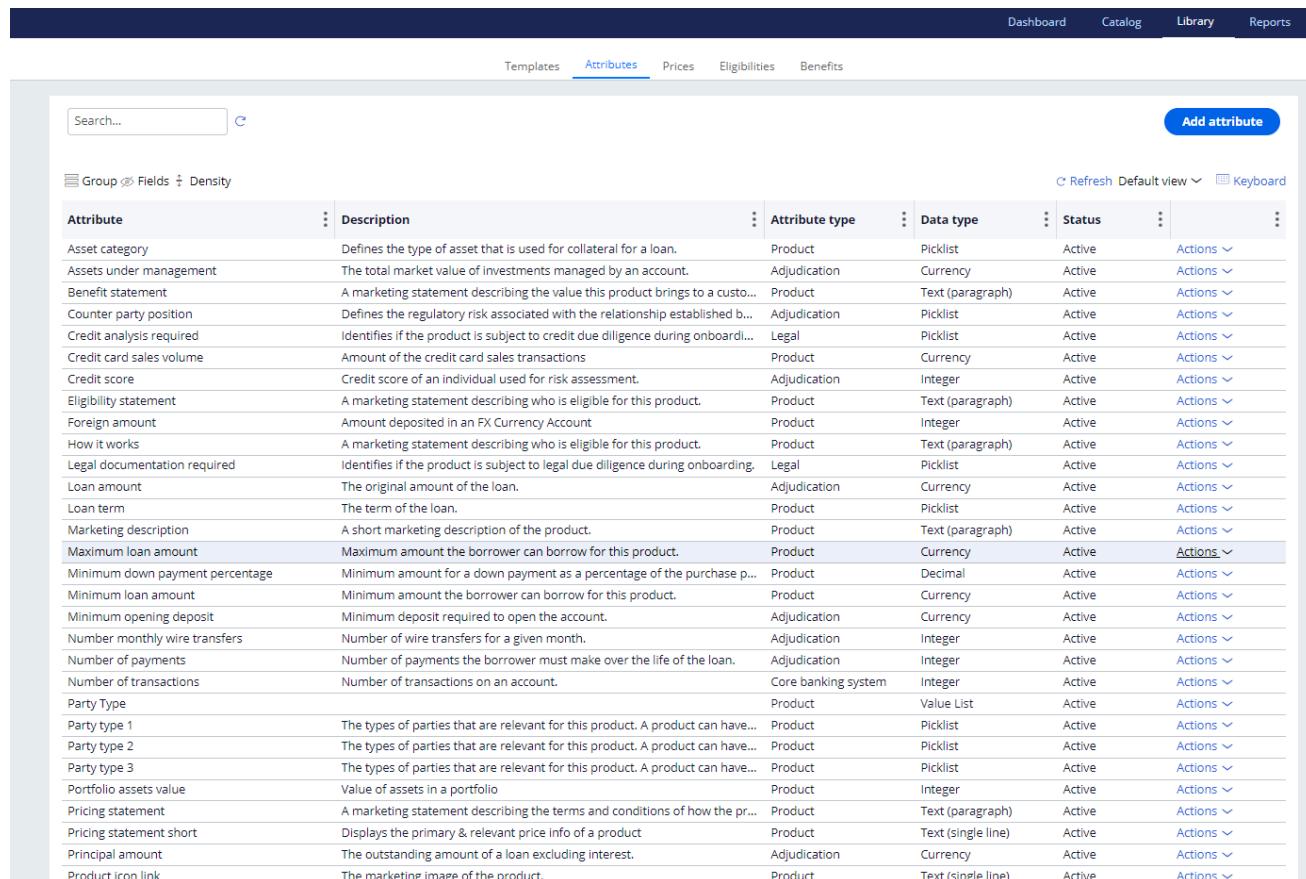
Product catalog ruleset

The Product Catalog Ruleset Dynamic System Setting defines which ruleset the products should be written to PDFS ships with this DSS setting set to *PegaPDFCatalog*.

This DSS setting can be set on the Financial Services > PDFS Configurations landing page. Make sure that this setting is set to *PegaPDFCatalogDemo*.

Attributes

There are a set of attributes that ship with PDFS and can be used directly in the creation of products. Review this list of attributes to see if there are any new attributes needed to support the products you are creating.



The screenshot shows the Pega Foundation for Financial Services '23 PDFS Configurations landing page. The top navigation bar includes links for Dashboard, Catalog, Library, and Reports. Below the navigation is a search bar and a blue 'Add attribute' button. The main content area is titled 'Attributes' and contains a table with columns for Attribute, Description, Attribute type, Data type, Status, and Actions. The table lists various attributes such as Asset category, Assets under management, Benefit statement, Counter party position, Credit analysis required, Credit card sales volume, Credit score, Eligibility statement, Foreign amount, How it works, Legal documentation required, Loan amount, Loan term, Marketing description, Maximum loan amount, Minimum down payment percentage, Minimum loan amount, Minimum opening deposit, Number monthly wire transfers, Number of payments, Number of transactions, Party Type, Party type 1, Party type 2, Party type 3, Portfolio assets value, Pricing statement, Pricing statement short, Principal amount, and Product icon link. The table rows have alternating background colors for readability.

Attribute	Description	Attribute type	Data type	Status	Actions
Asset category	Defines the type of asset that is used for collateral for a loan.	Product	Picklist	Active	Actions
Assets under management	The total market value of investments managed by an account.	Adjudication	Currency	Active	Actions
Benefit statement	A marketing statement describing the value this product brings to a customer.	Product	Text (paragraph)	Active	Actions
Counter party position	Defines the regulatory risk associated with the relationship established between a counterparty and the organization.	Adjudication	Picklist	Active	Actions
Credit analysis required	Identifies if the product is subject to credit due diligence during onboarding.	Legal	Picklist	Active	Actions
Credit card sales volume	Amount of the credit card sales transactions.	Product	Currency	Active	Actions
Credit score	Credit score of an individual used for risk assessment.	Adjudication	Integer	Active	Actions
Eligibility statement	A marketing statement describing who is eligible for this product.	Product	Text (paragraph)	Active	Actions
Foreign amount	Amount deposited in an FX Currency Account.	Product	Integer	Active	Actions
How it works	A marketing statement describing who is eligible for this product.	Product	Text (paragraph)	Active	Actions
Legal documentation required	Identifies if the product is subject to legal due diligence during onboarding.	Legal	Picklist	Active	Actions
Loan amount	The original amount of the loan.	Adjudication	Currency	Active	Actions
Loan term	The term of the loan.	Product	Picklist	Active	Actions
Marketing description	A short marketing description of the product.	Product	Text (paragraph)	Active	Actions
Maximum loan amount	Maximum amount the borrower can borrow for this product.	Product	Currency	Active	Actions
Minimum down payment percentage	Minimum amount for a down payment as a percentage of the purchase price.	Product	Decimal	Active	Actions
Minimum loan amount	Minimum amount the borrower can borrow for this product.	Product	Currency	Active	Actions
Minimum opening deposit	Minimum deposit required to open the account.	Adjudication	Currency	Active	Actions
Number monthly wire transfers	Number of wire transfers for a given month.	Adjudication	Integer	Active	Actions
Number of payments	Number of payments the borrower must make over the life of the loan.	Adjudication	Integer	Active	Actions
Number of transactions	Number of transactions on an account.	Core banking system	Integer	Active	Actions
Party Type		Product	Value List	Active	Actions
Party type 1	The types of parties that are relevant for this product. A product can have...	Product	Picklist	Active	Actions
Party type 2	The types of parties that are relevant for this product. A product can have...	Product	Picklist	Active	Actions
Party type 3	The types of parties that are relevant for this product. A product can have...	Product	Picklist	Active	Actions
Portfolio assets value	Value of assets in a portfolio.	Product	Integer	Active	Actions
Pricing statement	A marketing statement describing the terms and conditions of how the product is priced.	Product	Text (paragraph)	Active	Actions
Pricing statement short	Displays the primary & relevant price info of a product.	Product	Text (single line)	Active	Actions
Principal amount	The outstanding amount of a loan excluding interest.	Adjudication	Currency	Active	Actions
Product icon link	The marketing image of the product.	Product	Text (single line)	Active	Actions

Attributes

Price

You should also create the price rules that are required by your products, and there are a set of price rules in the PDFS sample application that you can use as models.

Follow these guidelines when creating price rules:

- Consider reusability - Make sure the price rule is high enough in the product structure, so it can be reused by other products. Don't always put it at the top, but put it at the node in the structure that makes the most sense.
- The currency should be blank. This will indicate that the default currency specified in the organization operational structure will be used at run-time.
- If you create a price rule that is a percentage of an attribute, make sure the attribute that you need has been created first.
- If the price rule will use a lookup table, build that first in Dev Studio and reference it here.

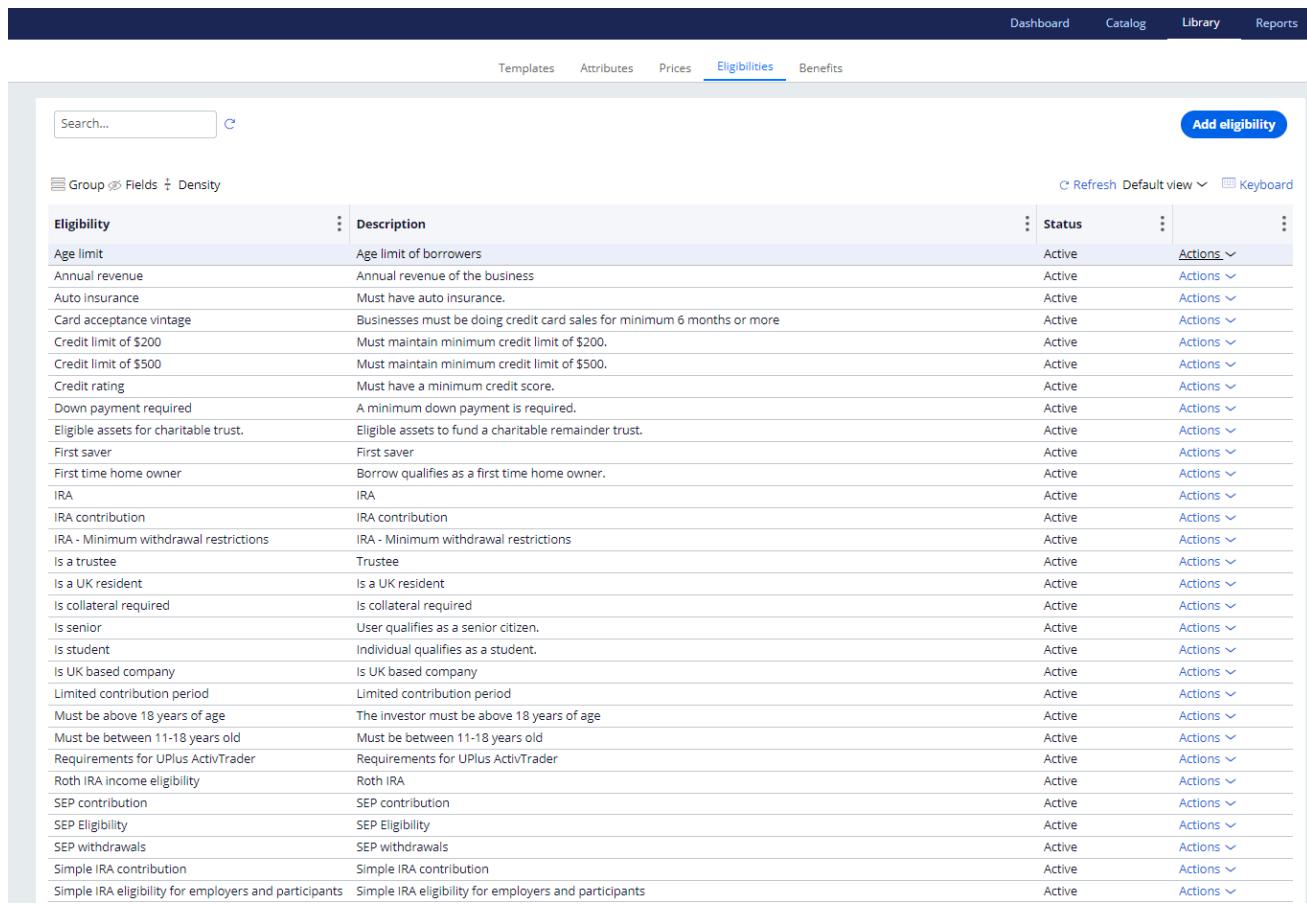
Group (1) Fields Density							Refresh	Default view	Keyboard
	Price	Description	Type	Value configuration	Currency	Status			
Category: Cards (Retail)							Total 4		
Balance transfer APR	Rate on moneys transferred from other cards	Rate	Type Value		Active	Actions			
Cash advance APR	Rate on cash advances	Rate	Type Value		Active	Actions			
Overdraft advance APR	Rate on overdraft advances	Rate	Type Value		Active	Actions			
Purchases APR	Rate charged for purchases on the card	Rate	Type Value		Active	Actions			
Category: Deposits (Retail)							Total 3		
APY	Annual Percentage Yield	Rate	Type Value		Active	Actions			
Overdraft fee	Fee charged when balance is less than 0.	Fee	Type Value		Active	Actions			
Service Fee	Recurring Service Fee	Fee	Type Value		Active	Actions			
Category: Lending (Retail)							Total 3		
Late payment fee	Amount charged if the payment is late.	Fee	Type Value		Active	Actions			
Origination fee	Fee charged by the lender to process the loan. Fee is...	Fee	Type Value		Active	Actions			
Prepayment penalty fee	Fee for prepayment of the loan.	Fee	Type Value		Active	Actions			
Category: Money Market (Retail>Deposits)							Total 1		
Early withdrawal penalty	Early withdrawal penalty	Fee	Type Value		Active	Actions			
Category: Mortgage (Retail>Lending>Secured)							Total 7		
APR	Annual percentage rate	Rate	Industry benchmark		Active	Actions			
5/1 ARM	5/1 Adjustable Rate Mortgage	Rate			Active	Actions			
Fixed interest rate	Fixed interest rate	Rate	External document		Active	Actions			
Flood certification fee	Cost of determining if property is in a flood zone.	Fee	Type Value		Active	Actions			
7/1 ARM	7/1 Adjustable Rate Mortgage	Rate			Active	Actions			
Tax service fee	Cost of verifying property taxes on the property.	Fee	Type Value		Active	Actions			
3/1 ARM	3/1 Adjustable Rate Mortgage	Rate			Active	Actions			
Category: Retail							Total 6		
Annual fee	Fee charged to the account annually.	Fee	Type Value		Active	Actions			
Foreign currency conversion	Fee to convert from one currency to another.	Fee	Type Value		Active	Actions			
Foreign transaction fee	3% of the U.S. Dollar amount of each transaction ma...	Fee	Type Value		Active	Actions			
Interest rate	The cost of the product.	Rate	Type Value		Active	Actions			
Monthly fee	Fee charged to account each month.	Fee	Type Value		Active	Actions			
Wire transfer fee	Cost of wiring funds	Fee	Lookup table		Active	Actions			

Price

Eligibility

Determine the eligibility requirements for your product. There are many eligibility rules already in the PDFS sample application.

The eligibility rules are not associated with the product structure, but are just a list of rules, so check to see if what you need has already been created and use that.



The screenshot shows a list of eligibility rules in a table format. The columns are: Eligibility, Description, Status, and Actions. The 'Actions' column contains a dropdown menu with options like 'Edit', 'Delete', and 'View'. The 'Status' column shows all entries as 'Active'. The 'Description' column provides a brief explanation for each rule, such as 'Age limit of borrowers' or 'Must have auto insurance'. The 'Eligibility' column lists the specific rule names.

Eligibility	Description	Status	Actions
Age limit	Age limit of borrowers	Active	Actions
Annual revenue	Annual revenue of the business	Active	Actions
Auto insurance	Must have auto insurance.	Active	Actions
Card acceptance vintage	Businesses must be doing credit card sales for minimum 6 months or more	Active	Actions
Credit limit of \$200	Must maintain minimum credit limit of \$200.	Active	Actions
Credit limit of \$500	Must maintain minimum credit limit of \$500.	Active	Actions
Credit rating	Must have a minimum credit score.	Active	Actions
Down payment required	A minimum down payment is required.	Active	Actions
Eligible assets for charitable trust.	Eligible assets to fund a charitable remainder trust.	Active	Actions
First saver	First saver	Active	Actions
First time home owner	Borrow qualifies as a first time home owner.	Active	Actions
IRA	IRA	Active	Actions
IRA contribution	IRA contribution	Active	Actions
IRA - Minimum withdrawal restrictions	IRA - Minimum withdrawal restrictions	Active	Actions
Is a trustee	Trustee	Active	Actions
Is a UK resident	Is a UK resident	Active	Actions
Is collateral required	Is collateral required	Active	Actions
Is senior	User qualifies as a senior citizen.	Active	Actions
Is student	Individual qualifies as a student.	Active	Actions
Is UK based company	Is UK Based company	Active	Actions
Limited contribution period	Limited contribution period	Active	Actions
Must be above 18 years of age	The investor must be above 18 years of age	Active	Actions
Must be between 11-18 years old	Must be between 11-18 years old	Active	Actions
Requirements for UPlus ActivTrader	Requirements for UPlus ActivTrader	Active	Actions
Roth IRA income eligibility	Roth IRA	Active	Actions
SEP contribution	SEP contribution	Active	Actions
SEP Eligibility	SEP Eligibility	Active	Actions
SEP withdrawals	SEP withdrawals	Active	Actions
Simple IRA contribution	Simple IRA contribution	Active	Actions
Simple IRA eligibility for employers and participants	Simple IRA eligibility for employers and participants	Active	Actions

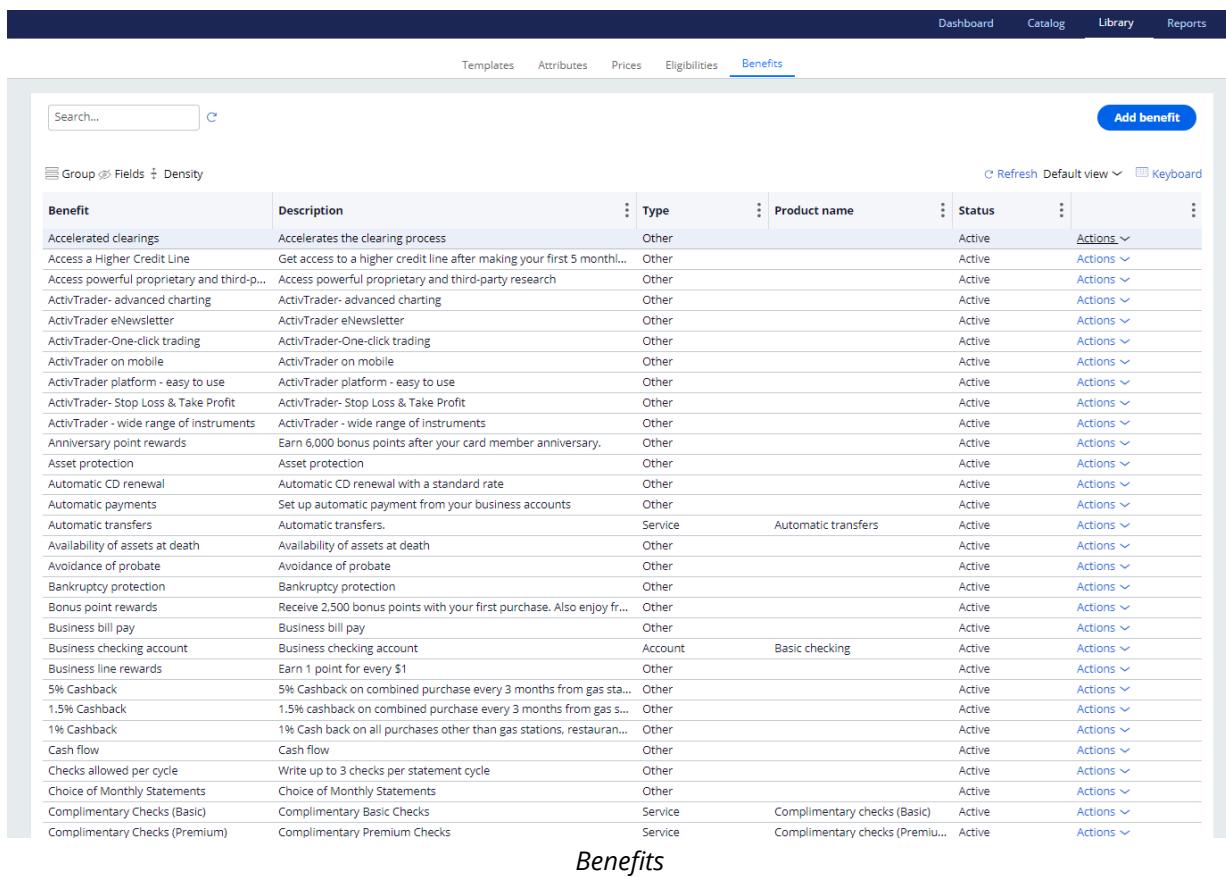
Eligibility

Benefits

Determine the benefits that are required for your product. There are many benefit rules already in the PDFS sample application. The benefit rules are not associated with the product structure, but are just a list of rules, so check to see if what you need has already been created and use that.

When creating benefits, we have been using sentence case to give a consistent look to the benefit list in the library. The elements of the benefit show up in different parts of the system, so choose the text you use accordingly:

- The Benefit Name – Used in the list of benefits when adding them to the template and product.
- The Benefit Description – Used by the consuming applications when they show a bulleted list of benefits.
- The Benefit Details – A more detailed description of what the benefit is and how it is used. Currently, this is not displayed on the screen in the consuming apps, but it could be in the future.



The screenshot shows a web-based application interface for managing benefits. At the top, there's a dark header bar with tabs for 'Dashboard', 'Catalog', 'Library' (which is currently selected), and 'Reports'. Below the header is a navigation bar with links for 'Templates', 'Attributes', 'Prices', 'Eligibilities', and 'Benefits'. A search bar is located at the top left, and an 'Add benefit' button is at the top right. The main content area is titled 'Benefits' and contains a table with the following columns: Benefit, Description, Type, Product name, Status, and Actions. The table lists numerous benefit entries, such as 'Accelerated clearings', 'Access a Higher Credit Line', and various ActivTrader features like 'advanced charting' and 'eNewsletter'. Each entry includes a brief description, its type (e.g., Other, Service, Account), the product it's associated with, its status (Active), and an 'Actions' dropdown menu.

Benefit	Description	Type	Product name	Status	Actions
Accelerated clearings	Accelerates the clearing process	Other		Active	Actions
Access a Higher Credit Line	Get access to a higher credit line after making your first 5 month...	Other		Active	Actions
Access powerful proprietary and third-p...	Access powerful proprietary and third-party research	Other		Active	Actions
ActivTrader- advanced charting	ActivTrader- advanced charting	Other		Active	Actions
ActivTrader eNewsletter	ActivTrader eNewsletter	Other		Active	Actions
ActivTrader-One-click trading	ActivTrader-One-click trading	Other		Active	Actions
ActivTrader on mobile	ActivTrader on mobile	Other		Active	Actions
ActivTrader platform - easy to use	ActivTrader platform - easy to use	Other		Active	Actions
ActivTrader- Stop Loss & Take Profit	ActivTrader- Stop Loss & Take Profit	Other		Active	Actions
ActivTrader - wide range of instruments	ActivTrader - wide range of instruments	Other		Active	Actions
Anniversary point rewards	Earn 6,000 bonus points after your card member anniversary.	Other		Active	Actions
Asset protection	Asset protection	Other		Active	Actions
Automatic CD renewal	Automatic CD renewal with a standard rate	Other		Active	Actions
Automatic payments	Set up automatic payment from your business accounts	Other		Active	Actions
Automatic transfers	Automatic transfers.	Service	Automatic transfers	Active	Actions
Availability of assets at death	Availability of assets at death	Other		Active	Actions
Avoidance of probate	Avoidance of probate	Other		Active	Actions
Bankruptcy protection	Bankruptcy protection	Other		Active	Actions
Bonus point rewards	Receive 2,500 bonus points with your first purchase. Also enjoy fr...	Other		Active	Actions
Business bill pay	Business bill pay	Other		Active	Actions
Business checking account	Business checking account	Account	Basic checking	Active	Actions
Business line rewards	Earn 1 point for every \$1	Other		Active	Actions
5% Cashback	5% Cashback on combined purchase every 3 months from gas sta...	Other		Active	Actions
1.5% Cashback	1.5% cashback on combined purchase every 3 months from gas s...	Other		Active	Actions
1% Cashback	1% Cash back on all purchases other than gas stations, restauran...	Other		Active	Actions
Cash flow	Cash flow	Other		Active	Actions
Checks allowed per cycle	Write up to 3 checks per statement cycle	Other		Active	Actions
Choice of Monthly Statements	Choice of Monthly Statements	Other		Active	Actions
Complimentary Checks (Basic)	Complimentary Basic Checks	Service	Complimentary checks (Basic)	Active	Actions
Complimentary Checks (Premium)	Complimentary Premium Checks	Service	Complimentary checks (Premiu...	Active	Actions

Benefits

Templates

The template describes the details and characteristics of a product. The more thought that goes into the template, the easier the creation of the actual products will be.

There are sets of attributes that are required for all products. These should be pulled into each template.

Attribute	Lock – required	Lock – read only	Default value	Reason it is required
Benefit statement	Yes	No		Needed by PMFS, SAFS
Eligibility statement	Yes	No		Needed by PMFS, SAFS
How it works	Yes	No		Needed by PMFS, SAFS
Pricing statement	Yes	No		Needed by PMFS, SAFS
Marketing description	Yes	No		Needed by PMFS, SAFS
Party type 1	Yes	No	Individual, Organization or Fund	Needed by PMFS, SAFS
Party type 2	No	No	Individual, Organization or Fund	Needed by PMFS, SAFS
Party type 3	No	No	Individual, Organization or Fund	Needed by PMFS, SAFS
Product icon link	No	No		Needed by PMFS, SAFS
Counter party position	Yes	Yes	Value appropriate for product	

Attribute	Lock – required	Lock – read only	Default value	Reason it is required
Credit analysis required	Yes	Yes	Value appropriate for product	
Legal documentation required	Yes	Yes	Value appropriate for product	
Vulnerable	Yes	Yes	Value appropriate for product	
Vulnerable reason	Yes	Yes	Value appropriate for product	

Attribute	Lock – required	Lock – read only	Default value	Reason it is required
Loan amount	Yes	No	None	Used by pricing, populated during onboarding
Principal amount	Yes	No	None	Used by pricing, populated during onboarding
Term	Yes	No	None	Used by pricing, populated

Attribute	Lock – required	Lock – read only	Default value	Reason it is required
				during onboarding
Credit score			None	Used by pricing, populated during onboarding
Minimum loan amount			None	Populated during onboarding
Maximum loan amount			None	Populated during onboarding
Minimum down payment percentage			None	Populated during onboarding
Number of payments			None	Used by core banking systems
Number of transactions			None	Used by core banking systems

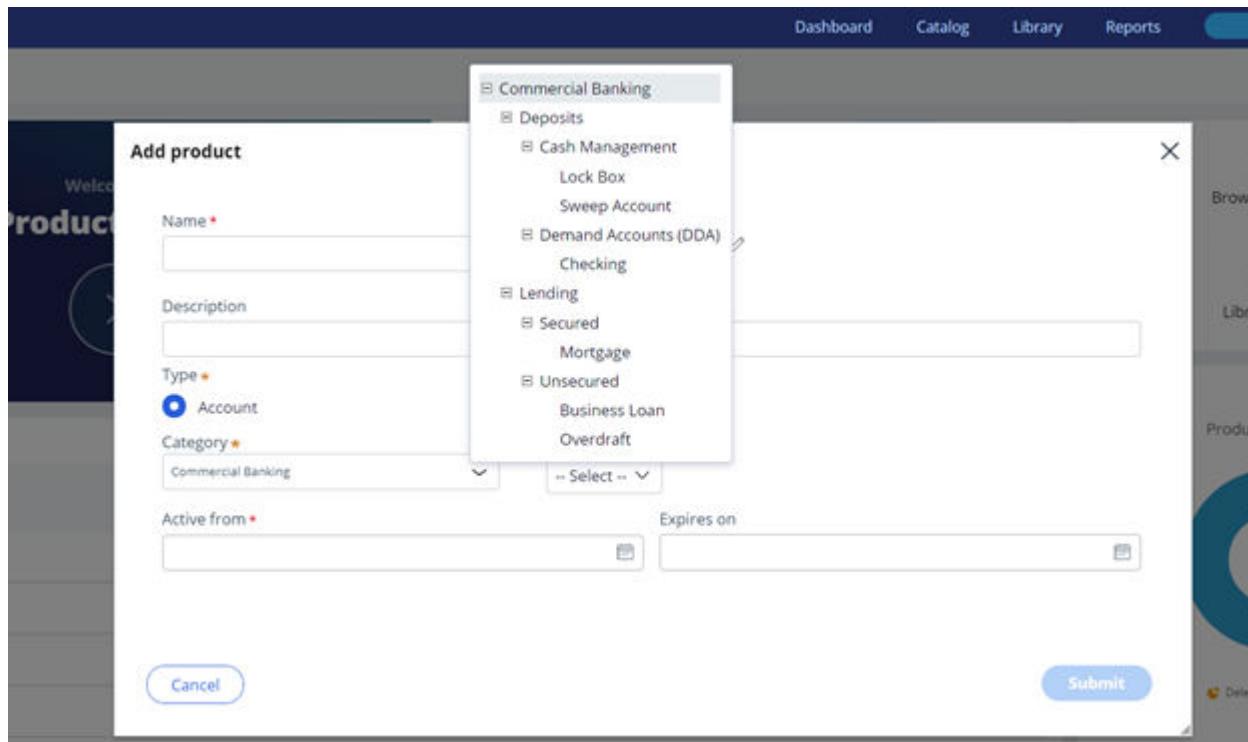
Attribute	Lock – required	Lock – read only	Default value	Reason it is required
Minimum opening deposit			None	Populated during onboarding

Attribute	Lock – required	Lock – read only	Default value	Reason it is required
Number monthly wire transfers			None	Used by pricing, Used by core banking systems
Number of transactions			None	Used by pricing, Used by core banking systems
Transaction amount			None	Used by core banking systems

Additional guidelines

Consider reusability - Make sure the template is high enough in the product structure so that it can be used to create the appropriate products. Do not always put it at the top, but put it at the node in the structure that makes the most sense.

The snapshot below shows the levels of the nodes you could be deciding to select for the template/product creation:

*Add product*

- When adding attributes to the template, they show up in the work area of the template and product in the order they are listed. You can rearrange the attributes in the list by dragging and dropping them in the order that you want them to show up on the screen. We have been grouping the attributes logically, listing the 'marketing attributes' that have a paragraph text box first, followed by other product types of attributes, followed by the attributes needed for onboarding, and then finally the attributes needed by CLM at the bottom. You can arrange them in the order that makes sense for your product.
- When adding default values to the template, remember that they can be changed in the product, so if the text is very similar for the different products, you can add the most common ones to the template and then just tweak the text in the resulting products.

This is where certain attributes are displayed in the WSS screens for Onboarding:

FX Volatility Protection

Optimized for corporate clients >\$500,000 annual revenues. One place for all local and international payment needs. Eliminates the need for multiple currency accounts and reconciliations for each. FX Volatility protection included.



Product name

Optimized for corporate clients >\$500,000 annual revenues. One place for all local and international payment needs. Eliminates the need for multiple currency accounts and reconciliations for each. FX Volatility protection included.

[Open Account](#)

[Request more info](#)

[Schedule an appointment](#)

Benefits

- ✓ Consolidate and manage foreign payments receivables more effectively
- ✓ View rates and trends to enable transaction execution when rates are favorable
- ✓ Conduct transactions in most currencies
- ✓ Employ hedging strategies for more volatile currencies.

Pricing

Select and tailor currency accounts to manage currency risk, assisted by experienced advisors. Transaction accounts - monthly fee per currency account with unlimited deposits and payment transaction optimized for your business. Savings accounts - minimum balances required or maximum transactions/month by currency risk tiers.

How it works

Individual accounts are created for each currency to allow rate and trend monitoring. Risk management and hedging strategies available for each account, with the guidance of experienced exchange advisors.

Eligibility

Product

Products

Products that are accounts are usually created in the lowest leaf node of the product structure. Products that are services can be higher in the product structure so that they can be reused when associating them with products. If most of the detail is populated in the template, creating the products is a very quick process.

Bundles

Once the products are created, they can be pulled into bundles.

Using an enrichment or screening marketplace component

Pega Foundation for Financial Services (PFFS) provides a single, unified approach for creating and extending integrations to a list of financial information providers that all Financial Services (FS) applications can access.

Enriching customer and company data from financial information providers enables financial services organizations many benefits including: operational efficiencies, more efficiently managing credit decisions, verifying identity, increasing fraud awareness,

providing more targeted and personalized cross-sell and upsell strategies, and optimizing collections treatments.

It is likely that you'll want to use several providers to automate customer journeys depending on the location, type of client, sort of data required, and agreements in place. Also, you'll want to take advantage of new services and information as providers make them available.

Supported Enrichment and Screening Data Providers (components):

- SWIFT KYC
- Avox
- IHS Markit
- Dun & Bradstreet
- Equifax
- Refinitiv's World Check One
- eIdentification



Note: The data structure in PFFS is extendable to support additional data providers as they are added.

- **Before you begin**
- **Implementation in PFFS**
- **Use case examples**

Before you begin

Downloading component from Pega Marketplace (formerly known as Pega Exchange):

- In the Enabled components section of the application rule, click on Manage components button.

Enabled components

+ Add component

Component	Version
No items	

[Manage components](#)

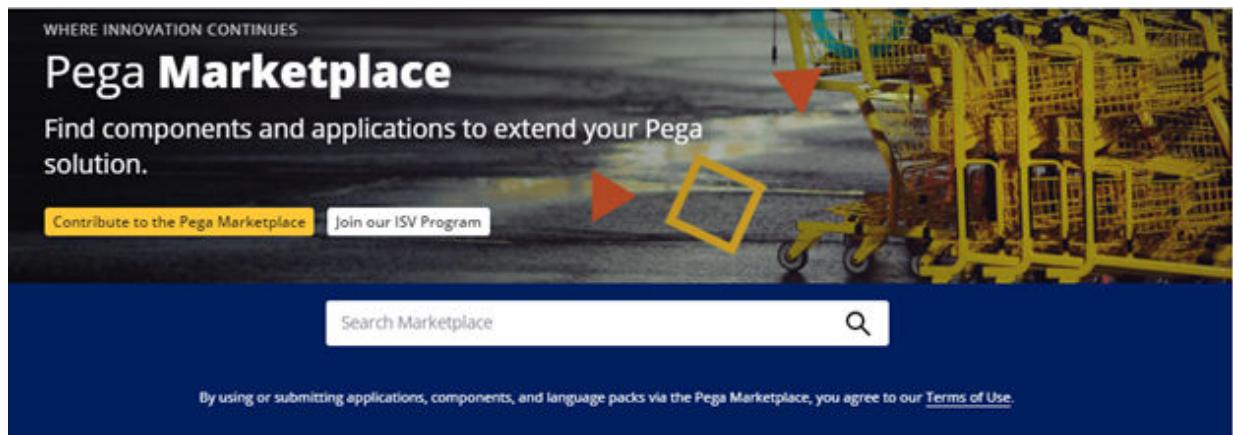
Manage Components

- If the desired component (For example, Avox) is already present on your system, check the Enabled box. Otherwise click on BROWSE APPS & COMPONENTS button to proceed to PEGA EXCHANGE (PEGA MARKETPLACE).

The screenshot shows the Pega Exchange interface. On the left, there's a search bar with placeholder text 'Search for components or apps...' and a 'Browse' button. Below it is a table titled 'Available components' with columns: Name, Version, Publisher, and Status. A single row is visible for 'Avox' version 08.04, published by 'Pega Foundation f...', with the 'Enabled' status checked. To the right of the table is a large circular diagram illustrating the process: 'Browse' leads to 'Download', which then leads to an 'Install' icon. At the bottom right of the diagram is a blue 'Install new' button.

Component Screen

- Search for the desired component.

*Marketplace*

Implementation in PFFS

PegaFS-Data-ExtProvider - Is a centralized class for making component calls and mapping returned data back into the Financial Services data structure.

Specialized child classes have been dedicated to handling enrichment and screening and output mapping.

Class: PegaFS-Data-ExtProvider-BusService-eEnrichment

Description: Class for retrieving search results and entity details for party enrichment.

Data Pages: Unified data pages for searching and entity details, selects applicable provider based on input, executes specialized data page for the provider to perform service call and handles output including errors.

D_Trigger_SearchEntity unified data page for searching with example call to Avox data provider.

Run Data Page: Trigger enrichment search
ID: D_Trigger_SearchEntity

Run context
Thread: Standard

Parameter	Value
EntityName	Pega
Provider	AVOX
MaxResults	5

Flush all instances of this data page before execution

Results

Property	Value												
pxObjClass	PegaFS-Data-ExtProvider-BusService-eEnrichment												
pzLoadTime	December 23, 2020 10:49:42 AM EST												
pzPageNameBase	D_Trigger_SearchEntity												
pzPageNameHash	_pa7568102242946296pz												
pxDPPParameters													
pxSourcePage													
ClientList													
ClientList[1]	<table border="1"> <tr> <td>Avid</td> <td>36146240</td> </tr> <tr> <td>FullName</td> <td>Pega Ag</td> </tr> <tr> <td>LegalEntityIdentifier</td> <td></td> </tr> <tr> <td>Provider</td> <td>AVOX</td> </tr> <tr> <td>TypeOfEntity</td> <td></td> </tr> <tr> <td>pxObjClass</td> <td>PegaFS-Data-ExtProvider-BusService-Client</td> </tr> </table>	Avid	36146240	FullName	Pega Ag	LegalEntityIdentifier		Provider	AVOX	TypeOfEntity		pxObjClass	PegaFS-Data-ExtProvider-BusService-Client
Avid	36146240												
FullName	Pega Ag												
LegalEntityIdentifier													
Provider	AVOX												
TypeOfEntity													
pxObjClass	PegaFS-Data-ExtProvider-BusService-Client												
Address													

Trigger search entity

D_Trigger_EntityDetails data page for entity details with example call to Avox provider.

Run Data Page: Trigger enrichment subscribed
D_Trigger_SubscribeEntity

Convert to test Clipboard Trace Run

Run context Thread Standard

Trigger enrichment subscribed was executed using: Standard Thread on RunRecordPrimaryPage page.

Parameter	Value
Provider	Avox
EntityID	36146240

Flush all instances of this data page before execution

Results

Property	Value
ErrorDescription	No technical infrastructure available for providerAvox
ResultCode	0
pxObjClass	PegaFS-Data-ExtProvider-BusService-eEnrichment
pzLoadTime	December 23, 2020 11:09:42 AM EST
pzPageNameBase	D_Trigger_SubscribeEntity
pzPageNameHash	_pa7569295676720806pz
ClientData	
Avid	36146240
CustomerStatus	Active
EntityClass	
EntityWebsite	
FullName	Pega Ag

Entity details

Class: PegaFS-Data-ExtProvider-BusService-eScreening

Description: Class for retrieving search results and entity details for party screening.

Data Page: D_Trigger_eScreening is a unified data page for retrieving entity screening details, selects applicable provider based on input, executes specialized data page for the provider to perform service call and handles output including errors.



Note: World Check One is the only screening provider currently enabled.

Class: PegaFS-Data-ExtProvider-BusService-Client

Description: Mapping returned data from enrichment and screening data page calls.

Data Transforms: Contains specialized data transforms dedicated to mapping response data from each data provider. Can be extended/overridden at consuming application layer to include additional properties.



Note: Additional child classes were created for distinct functionality for specific providers.

Usecase examples

Enrichment reports for organizations: Search and details case in PFFS.

External provider search (EXT-14001)

Get eEnrichment Reports

Avox ▾

Pega

Search

Customer name

Address

Legal entity identifier

Entity type

Provider

Pega Ag

Vaduz, LI

AVOX

Pega Capital Corp

AVOX

PEGA-VEL a.s

Krnov,
Moravskoslezsko, CZ

AVOX

PEGA Investments Sàrl

Fribourg, Fribourg, CH

506700LK16PC7K3ZBZ27

AVOX

Pega Medical Inc.

Laval, QC, CA

AVOX

Get eEnrichment Reports

[< Back to search](#)

PEGA Investments Sàrl Subscribed

AVOX • Last updated 27-Dec-2017

Entity status

Active

Headquarters address

Avenue De Beauregard 12,
Fribourg, CH
Avenue De Beauregard 12,Fribourg, FribourgCH,

Legal form

—

Regulatory identifier

—

Party details

Search screen

Reference

[Pega Marketplace](#)

Implementing case types

- [Defining the case type modifications and attributes](#)

Defining the case type modifications and attributes

A case is defined as a Service Case or a Service Request, which is a process that your staff will execute to service your customers. Each case contains one or more processes that will be extended or created to meet your business requirements. You can create new or edit existing case types based on the needs of your application during your planning process.

- [Updating case types](#)

If an existing case type is close to meeting your business requirements, you can modify it and use it.

- **Adding case types**

If you require a case type that is not like an existing case type, create a new one that meets your business requirements. If you want to add additional request types to your change request, create them as new case types.

- **Defining class group to database table mapping**

To define how your cases are stored in the database, create a relationship between the class and the physical database table where the class instances will be stored. Work with the database administrator at your site on this task.

- **Configuring case type locking**

For each case type, select whether only one user at a time can edit a case, or whether multiple users can concurrently edit the same case.

For more information on defining case-types, please see [Creating case types](#).

Pega Foundation for Financial Services includes sample cases for testing and demo purposes. You can view a list of the case types from the Designer Studio by selecting Case types icon from the left navigation panel.

Case Prefix	Case Description	Applies to This Class
R-	Requirement: Collect and validate document(s).	PegaFS-Work-Requirement
AA-	StartLoanProcess: Apply for personal loan	PegaFS-Work-NewAccount
A-	CreateAccounting: Create a general ledger	PegaAcct-Work-Sample

Case Prefix	Case Description	Applies to This Class
C-	CustomerSearch: Search and manage individuals and organizations.	PegaFS-Work-CustomerSearch
CS-	CustomerAssesment: Credit profile from Equifax.	PegaFS-Work-CustAssessment
H-	CreateHousehold	PegaFS-Work-Household
CK-	CustomerMarkitHistory: Entity data enrichment using MarkIt.	PegaFS-Work-CustAssessment-MarkIt
DB-	DBReporting: Resolve company address data from Dun and Bradstreet.	PegaFS-Work-DBReporting
EXT-	ExternalProviderSearch: Entity enrichment data from external providers.	PegaFS-Work-ExtProviderSearch
G-	Goal: Create and manage financial goals.	PegaFS-Work-Goal



Note: Customer Search case is described in the building features portion of this document.

PDFS

- [Product Designer for Financial Services installation guide](#)
- [Product Designer for Financial Services update guide](#)
- [Product Designer for Financial Services implementation guide](#)

Product Designer for Financial Services installation guide

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

Completing the prerequisite tasks

Before you install Product Designer for Financial Services, complete the following tasks.

1. Before starting an installationupdate, and before backing up your system, review the database policies and application permissions that are used by your installationupdate. Determine whether the application is permitted to update the database automatically or if you must generate the database scripts that your organization will use to manually make schema changes.
2. Install the latest available release of Pega Platform '23.
For more information, see the *Pega Platform Install Guide* for your environment available on the [Deploy Pega Platform](#) page.
3. Apply any required hotfixes to the Pega Platform before updating the Pega Product Designer for Financial Services application. For a list of required hotfixes for Pega Platform, see the [Pega Foundation for Financial Services Hotfixes](#).
4. Update the online help to the latest version from the Online Help Files section on the Pega Platform Update page. After you download the latest prhelp.war file,

deploy it in your installation before continuing. For deployment information, see the [Pega Platform Installation Guide for your application server](#).

5. Install and verify Pega Foundation for Financial Services '23

For more information, see the *Pega Foundation for Financial Services Installation Guide* on the [Pega Foundation for Financial Services product page](#).

6. Apply any required hotfixes for this application.

For the list of required hotfixes, see the [Pega Foundation for Financial Services Hotfixes](#) page.

7. Obtain the latest Product Designer for Financial Services application bundle. Apply any required hotfixes.

For the list of required Pega Product Designer for Financial Services hotfixes, see the [Pega Foundation for Financial Services Hotfixes](#).

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`
 - `pegarules.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 Product Designer for Financial Services, complete the following tasks.

Before you begin:

Make sure that you back up your system and that you complete all the prerequisite tasks. For more information, see [Completing the prerequisite tasks](#) and [Backing up your system](#).

Importing the application bundle

Product Designer for Financial Services '23 assets are located on the Pega Foundation for Financial Services distribution media, in the `\ResourceKit\PDFS` directory. Navigate to this directory to import the required files. Use the Import wizard to import the system data and rulesets for Product Designer for Financial Services to Pega Platform.



Note: Product Designer for Financial Services does not support Multi-Tenant environment.

1. Log in to Pega Platform at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.

For example:

administrator@pega.com

If this is the first time that you are logging in to this Pega Platform server after the installation, you are prompted to change the password.

2. In the header of Dev Studio, click **Configure > Application > Distribution > Import** and perform the following actions:
 - a. Depending on your web browser, click **Browse or Choose File**, select the **\ResourceKit\PDFS\Rules\PegaPDFS.jar** from your distribution image, and then click **Next**.
 - b. Ensure that you import all components by leaving the **Enable advanced mode to provide granular control over the import process** check box cleared, and then click **Next**.

Result:

The wizard identifies differences between the database schema of your current system and the schema of Product Designer for Financial Services '23.

- c. Based on the review of the database policies and application permissions that are used by your Pega Platform Installation Guide Prerequisites, select **Automatic** or **Manual**, and then click **Next**.
If you select **Manual**, see [Viewing and applying schema changes](#).
 - d. Follow the instructions on the screen, and when the import is completed, click **Done**.
3. Verify the total error count. If there are no errors, click **Done**.
 4. Repeat step 2 for the **ResourceKit\PDFS\rules\PegaPDFS_Implementation.jar** file from your distribution media.
You can now optionally install a PDFS demo application and its corresponding sample data as described in [Optional: Installing sample data and operators](#).

5. **Optional:** For the best performance and user experience, pre-assemble the rules in the application by running the Static Assembler utility.
For more information, see [Preassembling rules in an application by using the Static Assembler utility](#).
6. Resolve errors if any.
If you are unable to resolve them, post your questions to the [Pega Product Support Community](#).
7. Apply any required hotfixes. For the list of required Pega Product Designer for Financial Services hotfixes, see the [Pega Foundation for Financial Services Hotfixes](#).
8. In the **Explorer** panel, click Records > Application Definition > Application, and then verify that the *PegaPDFS* ruleset appears on the list.

Reset Unique Ids

For seamless creation of Templates, Products or Bundles, we recommend you run the utility “Reset Unique ID’s” to reset the work object ID’s post update. Go to Configure > Financial services > Sample Data > Reset Unique ID’s

The screenshot shows the Pega Platform interface with the following navigation path:

- Home
- Application: Product Designer for Financial Services 8.7 - Implementation
- Configure
- Search all options...
- Application
- Case Management
- Decisioning
- Data Model
- Mobile
- Reporting
- Integration
- Org & Security
- System
- User Interface
- Financial Services
- Configure CMS Preferences
- Equifax Service
- Operational Structure
- Product Structure
- Sample Data
- Snapshot Management
- PDFS Configurations
- Date Shift
- Reset Unique ID's**

Below the navigation bar, there are sections for "Guardrail warnings (last 7 days)" and "Security status".

	Severe	Moderate	Informational
Introduced by you	0	0	0
Introduced by team	0	0	0

And click on “Reset Unique ID’s” button.

The screenshot shows a web browser window with the URL 10.225.95.42:9080/prweb/app/PegaPDFSImplementation/_DvRH7K5NUqtHlyNPX/dzMWeR84qFve*/ISTANDARD?pzPostData=-1721470151. The page title is "Reset Unique ID's". The content area contains the following text:

Please click on button to reset the Unique IDs post upgrade for the indexes provided in the DSS **IndexesToResetUniqueId**

Please ensure to stay in the appropriate application context before the action is performed

Note: It is recommended to run this utility post update as the utility reserves a unique ID for templates, products or bundles which are going to be created

- ⓘ newly. Otherwise, you may end up experiencing an error saying “work object already exists” because work objects that were created earlier already exist in DB tables.

Unlock a new ruleset version

Since templates, products and bundles are custom rules, it is mandatory for us to have an open ruleset version of your choice to have these work objects created.

When creating the rules required to build products, the system writes these rules to a ruleset defined by the **Product Catalog Ruleset** Dynamic System Setting (DSS). Product Designer for Financial Services includes the PegaPDFSCatalog ruleset to implement the product catalog. This ruleset must be unlocked so the system can write rules to it, or you can input ruleset version of your choice in the DSS and have it unlocked.

Optional: Installing sample data and operators

The Product Designer for Financial Services application includes a demo application and sample customer data that you can use to explore the application's capabilities.



Note: Product Designer for Financial Services does not support Multi-Tenant environment.

At this stage of the process, you must manually import the following files from your distribution image:

- \ResourceKit\PDFS\SampleData\PegaPDFSSampleProducts.jar (Shared layer)
- \ResourceKit\PDFS\SampleData\PegaPDFSSampleObjects.jar (Shared layer)

1. Log in to at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.

For example:

`administrator@pega.com`

2. In the header of Dev Studio, click Configure > Application > Distribution > Import and perform the following actions:
 - a. Depending on your web browser, click Browse or Choose File, select each file from your distribution image, and then click Next.
 - b. Follow the instructions on the screen, and when the import is completed, click Done.

What to do next:

To prevent overwriting the production environment's *PegaPDFSCatalog* product catalog ruleset with rules created in the demo application, update *ProductCatalogRuleset* value to the specific ruleset used by the demo application.

Optional: Enabling sample operator accounts

Product Designer for Financial Services includes several sample operator accounts that are disabled for security purposes. Before you can use the sample application, enable a sample operator account.

1. Log in to Pega Platform at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.
2. In the header of Dev Studio, click Configure > Org & Security > Authentication > Operator Access.
3. In the **Disabled operators** section, select the check box next to the operator ID that you want to enable, and then click Enable selected.
4. In the confirmation dialog box, click Submit.

Result:

The **Enable operator** window appears with sample operators temporary passwords.

5. Click OK to close the dialog box.
When you log in by using the sample operator role the first time, you are prompted to reset your temporary password.

Product Designer for Financial Services update guide

- Completing the prerequisite tasks

- Backing up your system

Completing the prerequisite tasks

Before you update to the latest release of Product Designer for Financial Services, complete the following tasks.

1. Before starting an installationupdate, and before backing up your system, review the database policies and application permissions that are used by your installationupdate. Determine whether the application is permitted to update the database automatically or if you must generate the database scripts that your organization will use to manually make schema changes.
2. update to the latest available release of Pega Platform '23.
For more information, see the *Pega Platform update Guide* for your environment available on the [Deploy Pega Platform page](#).
3. Apply any required hotfixes to the Pega Platform before updating the Pega Product Designer for Financial Services application. For a list of required hotfixes for Pega Platform, see the [Pega Foundation for Financial Services Hotfixes](#).
4. Update the online help to the latest version from the Online Help Files section on the Pega Platform Update page. After you download the latest prhelp.war file, deploy it in your installation before continuing. For deployment information, see the [Pega Platform Installation Guide for your application server](#).
5. Install and verify Pega Foundation for Financial Services '23
For more information, see the *Pega Foundation for Financial Services Installation Guide* on the [Pega Foundation for Financial Services product page](#).
6. Apply any required hotfixes for this application.
For the list of required hotfixes, see the [Pega Foundation for Financial Services Hotfixes](#) page.
7. Obtain the latest Product Designer for Financial Services application bundle. Apply any required hotfixes.
For the list of required Pega Product Designer for Financial Services hotfixes, see the [Pega Foundation for Financial Services Hotfixes](#).

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
- pegarules.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 Product Designer for Financial Services from an earlier release, complete the following tasks.

Before you begin:

Make sure that you back up your system and that you complete all the prerequisite tasks. For more information, see [Completing the prerequisite tasks](#) and [Backing up your system](#).

- [Importing the application bundle](#)
- [Optional: Updating sample data and operators](#)
- [Optional: Enabling sample operator accounts](#)

Importing the application bundle

Product Designer for Financial Services '23 assets are located on the Pega Foundation for Financial Services distribution media, in the \ResourceKit\PDFS directory. Navigate to this directory to import the required files. Use the Import wizard to import the system data and rulesets for Product Designer for Financial Services to Pega Platform.



Note: Product Designer for Financial Services does not support Multi-Tenant environment.

1. Log in to Pega Platform at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.

**For example:**

`administrator@pega.com`

If this is the first time that you are logging in to this Pega Platform server after the installation, you are prompted to change the password.

2. In the header of Dev Studio, click **Configure > Application > Distribution > Import** and perform the following actions:
 - a. Depending on your web browser, click **Browse or Choose File**, select the `\ResourceKit\PDFS\Rules\PegaPDFS.jar` from your distribution image, and then click **Next**.
 - b. Ensure that you import all components by leaving the **Enable advanced mode to provide granular control over the import process** check box cleared, and then click **Next**.

Result:

The wizard identifies differences between the database schema of your current system and the schema of Product Designer for Financial Services '23.

- c. Based on the review of the database policies and application permissions that are used by your Pega Platform Update Guide Prerequisites, select **Automatic** or **Manual**, and then click **Next**.
If you select **Manual**, see [Viewing and applying schema changes](#).
- d. Follow the instructions on the screen, and when the import is completed, click **Done**.
3. Verify the total error count. If there are no errors, click **Done**.
4. Repeat step 2 for the `ResourceKit\PDFS\rules\PegaPDFS_Implementation.jar` file from your distribution media.
You can now optionally install a PDFS demo application and its corresponding sample data as described in [Optional: Installing sample data and operators](#).
5. **Optional:** For the best performance and user experience, pre-assemble the rules in the application by running the Static Assembler utility.
For more information, see [Preassembling rules in an application by using the Static Assembler utility](#).
6. Resolve errors if any.

If you are unable to resolve them, post your questions to the [Pega Product Support Community](#).

7. Apply any required hotfixes. For the list of required Pega Product Designer for Financial Services hotfixes, see the [Pega Foundation for Financial Services Hotfixes](#).
8. In the **Explorer** panel, click Records > Application Definition > Application, and then verify that the *PegaPDFS* ruleset appears on the list.

Result:

The application bundle import is complete. Note that updating does not affect existing enabled operators.

Reset Unique Ids

For seamless creation of Templates, Products or Bundles, we recommend you run the utility “Reset Unique ID’s” to reset the work object ID’s post update. Go to Configure > Financial services > Sample Data > Reset Unique ID’s

The screenshot shows the Pega Platform Dev Studio interface. The top navigation bar includes links for 'GCS pega Interact...', 'SWD/PSOR User Po...', 'Lab owner', 'What's new in Pega...', 'maintained instance', 'pffs dev 8.7', 'pffs build 8.7', 'QA 8.7', 'MultiApp instance', 'RemoteCase type c...', 'ATQ', 'PDC', 'DCO Testing', 'Incident Portal', and 'ManageStandardEligibility'. The main menu on the left has sections like 'Application', 'Case Management', 'Deciding', etc., with 'Financial Services' expanded to show 'Operational Structure', 'Product Structure', 'Sample Data', 'Snapshot Management', 'PDFS Configurations', and 'Reset Unique ID's'. A central dashboard displays 'Guardrail warnings (last 7 days)' with two rows: 'Introduced by you' and 'Introduced by team', each with three columns: 'Severe', 'Moderate', and 'Informational', all showing '0' counts. To the right, there is a 'Security status' section stating 'Security guide not configured'.

And click on “Reset Unique ID’s” button.

The screenshot shows a web browser window with the URL 10.225.95.42:9080/prweb/app/PegaPDFSImplementation/_DVrRH7K5NUqtHlyNPX/dzMWeR84qFve*/ISTANDARD?pzPostData=-1721470151. The page title is "Reset Unique ID's". The content area contains the following text:

Please click on button to reset the Unique IDs post upgrade for the indexes provided in the DSS **IndexesToResetUniqueId**

Please ensure to stay in the appropriate application context before the action is performed

Note: It is recommended to run this utility post update as the utility reserves a unique ID for templates, products or bundles which are going to be created

- ⓘ newly. Otherwise, you may end up experiencing an error saying “work object already exists” because work objects that were created earlier already exist in DB tables.

Unlock a new ruleset version

Since templates, products and bundles are custom rules, it is mandatory for us to have an open ruleset version of your choice to have these work objects created.

When creating the rules required to build products, the system writes these rules to a ruleset defined by the **Product Catalog Ruleset** Dynamic System Setting (DSS). Product Designer for Financial Services includes the PegaPDFSCatalog ruleset to implement the product catalog. This ruleset must be unlocked so the system can write rules to it, or you can input ruleset version of your choice in the DSS and have it unlocked.

Optional: Updating sample data and operators

The Product Designer for Financial Services application includes a demo application and sample customer data that you can use to explore the application's capabilities.



Note: Product Designer for Financial Services does not support Multi-Tenant environment.

At this stage of the process, you must manually import the following files from your distribution image:

- \ResourceKit\PDFS\SampleData\PegaPDFSSampleProducts.jar (Shared layer)
- \ResourceKit\PDFS\SampleData\PegaPDFSSampleObjects.jar (Shared layer)

1. Log in to at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.



For example:

administrator@pega.com

2. In the header of Dev Studio, click Configure > Application > Distribution > Import and perform the following actions:
 - a. Depending on your web browser, click Browse or Choose File, select each file from your distribution image, and then click Next.
 - b. Follow the instructions on the screen, and when the import is completed, click Done.

What to do next:

To prevent overwriting the production environment's *PegaPDFSCatalog* product catalog ruleset with rules created in the demo application, update *ProductCatalogRuleset* value to the specific ruleset used by the demo application.

Optional: Enabling sample operator accounts

Product Designer for Financial Services includes several sample operator accounts that are disabled for security purposes. Before you can use the sample application, enable a sample operator account.

1. Log in to Pega Platform at `https://hostname:port/prweb` with the administrative ID and password that you set up during the installation of Pega Platform.
2. In the header of Dev Studio, click Configure > Org & Security > Authentication > Operator Access.
3. In the **Disabled operators** section, select the check box next to the operator ID that you want to enable, and then click Enable selected.
4. In the confirmation dialog box, click Submit.

Result:

The **Enable operator** window appears with sample operators temporary passwords.

5. Click OK to close the dialog box.
When you log in by using the sample operator role the first time, you are prompted to reset your temporary password.

Product Designer for Financial Services implementation guide

- Application overview

- **Initiate stage**
- **Delivery stage**
- **Production maintenance and monitoring**

Application overview

Pega Product Designer for Financial Services provides product definitions used by other Pega Financial Services applications along with an example of what a Pega product catalog application could look like.

If a customer's product catalog system of record will be used, please refer to the Foundation for Financial Services product page for product catalog classes and data pages. This implementation guide describes the example Pega product catalog.

Pega Product Designer for Financial Services provides an intuitive UI that enables product designers to develop the products without having the need of technical knowledge to build the sections, flow actions, activities, etc. Implementing an application like this would accelerate the process of creating new products at financial institutions with less effort. Product designers can build new financial products for their institutions using its capabilities, which include:

- Creating library components
- Managing product catalogs
- Introducing life-cycle processes for template/product and bundling
- Managing reports

Note: Pega Product Designer for Financial Services does not currently

- ⓘ support multitenant environments due to it using shared rulesets for products, pricing, eligibility, and other product-related attributes.

Implementation delivery methodology

Pegasystems recommends the use of an Agile delivery model for a Pega application implementation. Usually customers report that the Scrum-based standard delivery model is well-suited for implementations. In rare cases where a more Waterfall based implementation methodology is better suited, Pegasystems recommends using an iterative, waterfall delivery process. These two proven implementation methodologies can facilitate the breakdown and delivery of incremental processes changes into production environments. The Pegasystems delivery approach has two primary stages:

- Initiate stage
- Delivery stage

Initiate stage

In the Initiate stage, the implementation teams build out the foundation or baseline of the application and prepare for the work that is necessary to configure the first production release and subsequent extension releases. Building a strong foundation to support expansion and reuse is core to the success of an implementation.

The Initiate stage can be divided into three sub-stages:

- Plan: Align the vision and roadmap to establish the foundation for the implementation, which includes:
 - Defining the production release milestone
 - Refining scope alignment
 - Determining the implementation methodology
- Setup: Validate and review the Pega-provided features and capabilities with customer requirements and expectations, which includes:
 - Establishing environments and processes
 - Creating the baseline application
 - Loading sample data
 - Demonstrating the baseline application
 - Performing a gap analysis

- Reviewing business needs and outcomes
- Prepare: Prepare for and implement the final Product Designer application into the organization. Integration is beyond the scope of this guide, which does not discuss integration tasks, which includes:
 - Confirming resources
 - Enabling team members
 - Establishing governance

At the end of the Initiate stage, the scope of the work for the first production release is defined. For scrum deliveries, the scope of the work is identified with an initial backlog; for an iterative waterfall delivery, a specification list and schedule are prepared and finalized.

Delivery stage

The Delivery stage is dependent on the delivery methodology selected during the Initiate stage. During this stage, the implementation team designs, builds, configures, and tests the application by using its selected implementation methodology (scrum or iterative waterfall). This guide provides information to support both implementation methodologies during the delivery stage.

The goal of the Delivery stage is to implement a fully tested and performance-tuned application into production environments. This goal is accomplished by completing the tasks described in this document by organizing the Application Feature backlog so that the delivery team can configure the application incrementally.

Initiate stage

The Initiate stage includes the tasks:

Creating the application

Before beginning, check to ensure that the Product Designer for Financial Services application is installed. The Product Designer for Financial Services application is a

complete application that is included on the Pega Foundation for Financial Services media.

See the [Product Designer for Financial Services Installation Guide](#) on the Pega Foundation for Financial Services product page.

The Product Designer for Financial Services installation includes the installation of an implementation application, which you can use directly with no changes required; however, Pega allows and provides functionality for advanced users to create a customized version of the implementation application. To do so, copy the PegaPDFSImplementation application rule, configure it with site-specific custom rulesets, and save it appropriately.

When creating the rules required to build products, the system writes these rules to a ruleset defined by the **Product Catalog Ruleset** Dynamic System Setting (DSS). Product Designer for Financial Services includes the PegaPDFSCatalog ruleset to implement the product catalog. This ruleset must be unlocked so the system can write rules to it. The PegaPDFSImplementation application should have an unlocked version of this ruleset.

To ensure that there is an unlocked version of the PegaPDFSCatalog ruleset:

1. Log in to the application as **PDFSSysAdmin** using the password that you specified when you enabled this operator; to enable Pega-provided operators, see [Enabling operators](#).
2. Click Dev Studio > Application > Structure > Ruleset Stack.
3. Click the PegaPDFSCatalog ruleset in the PegaPDFSImplementation application to view the ruleset versions. If there is no unlocked version, create a new unlocked version by clicking “+”.

To verify that the Product Catalog Ruleset DSS is set to PegaPDFSCatalog:

4. Click Dev Studio > Financial Services > PDFS Configurations.
5. Verify the **ProductCatalog Ruleset** setting is set to **PegaPDFSCatalog**.

Loading customer-supplied sample data

Pega recommends that you load customer-supplied sample data so that implementation efforts are more meaningful for the client, however sample data is provided with the Pega Foundation for Financial Services application that supports the demonstration of this application along with the other Financial Services applications.

Pega Foundation for Financial Services also comes with a sample application containing sample data that can demonstrate the features and functions of Product Designer for Financial Services.

To install the PDFSDemo application, follow the instructions provided in the *Product Designer for Financial Services Installation Guide* available on the Pega Foundation for Financial Services .

Reviewing artifacts from the gap analysis

During the Solution Implementation Gap Analysis, several requirements, specifications, and other artifacts are captured to represent the scope of the work that is targeted for the first production release and the subsequent extended production releases. If captured in the Pega Platform 8.4, these artifacts can be reused to streamline the first production release scope identification.

You can export application specification and requirements artifacts from Pega Platform, and then import them into your new baseline application so that you can build on the content that has already been created. For more information about exporting and importing these documents, see Exporting specifications and requirements and Importing specifications and requirements.

Follow these steps to generate the application document.

Application profile

This document is created during the Sales stage and includes the process stages and steps, case types, reports, requirements, specifications, participants, collaborators, and

actors that are associated with the implementation of your application. It is a long document that completely defines the scope of the entire application, and it also includes the results of the presale gap analysis.

When you generate this document, you have the option to include process flow diagrams. Follow these steps to generate the application profile:

1. Log in to the application as **PDFSSysAdmin** using the password that you specified when you enabled this operator; to enable Pega-provided operators, see [Enabling operators](#).
2. Click Dev Studio > Application > Tools > Document.
3. In the Select a document type section, click **Application profile**.
4. In the Application field, select Product Designer for Financial Services. This value determines the rulesets for which the Application Profile is generated.
5. Enter values in the Project Name and Organization Name fields.
6. Click Generate document.

Application document

This document and its associated controls provide you with the option to customize the presentation of the application information based on the document's intended audience. When you run the Application Document tool, you can include or exclude content by selecting and clearing various options.

During an implementation, this document reflects the current state of your application. The Application Document tool automatically updates the content of your document as you modify and extend your application to reflect additional requirements. If you have not yet built anything, the content of this document is the same as the application profile.

Follow these steps to generate the application document:

1. Log in to the application as **PDFSSysAdmin** using the password that you specified when you enabled this operator; to enable Pega-provided operators, see [Enabling operators](#).
2. Click Dev Studio > Application > Tools > Document.
3. In the **Select a document type** section, click **Application** document.
4. Click **Generate document**.

You can control the level of detail included in the Application document by modifying the settings in the **Options** section.

Specification document

This document shows all the specifications and linked requirements associated with your application at the time that you generate the document. You can generate this document at any point during the implementation life cycle to include any updates to the application's specifications and requirements.

Review this document before and after each sprint, using the available filters to tailor the content to meet the business requirements of your implementation.

Follow these steps to generate the specification document:

1. Log in to the application as **PDFSSysAdmin** using the password that you specified when you enabled this operator; to enable Pega-provided operators, see [Enabling operators](#).
2. Click **Dev Studio > Application > Tools > Document** to open the Document Application tool.
3. Click **Specification document**.
4. In the Application field, select an application in your stack.
5. Select the Include built-on application layer specifications check box to include specifications from the selected application and specifications from any application in the stack, excluding the Pega layers such as PegaRULES and PegaDM. This check box is available if your application has a built-on application.

6. Configure the content and appearance of the document by selecting the following options:
 - **Change log** - Includes a table of history memos for each specification.
 - **Acceptance criteria** - Includes acceptance criteria for each specification.
Results depend on the methodology used by your project.
 - **Linked requirements** - Includes the requirements that are linked to each specification.
7. Select the check boxes for the specifications that you want to include. All of the available specifications for your application are listed.
8. You might want to include only certain specifications if you are generating a document for a specific case type. You can filter the list of specifications by using the options in each column header. Attachments are included in the generated document when you select the **Include in document?** check box on the Application Specification rule form.
9. Specify the order of chapters for the generated document.
10. Optional: Click **Save document configuration** from the Actions menu to save your configuration settings.
11. Click **Generate document** to generate the Word document with the current settings.

Delivery stage

The Delivery stage includes the tasks:

Defining requirements

Use a structured approach in your implementation methodology in order to maintain the goals and schedule of your application deployment. To ensure success, determine the key aspects of the design that will affect the behavior of your application by defining your application requirements:

- Define the security model and product operational structure
- Configure default system settings
- Define the product components

Defining the security model and product operational structure

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

- Authentication: proves to the application that you are who you say you are. Pega applications support several authentication protocols.
- Authorization: tells the application which functions you may perform. Pega applications define this using access group and role configurations.

Pega security planning additionally involves configuring the application's operational structure and operator attributes

Pega applications provide organization the ability to fine-tune security control through using access settings and denial rules in the application. Many integration rules also incorporate authentication.

Defining the security model includes these tasks:

- Supported authentication schemes
- Defining an authentication scheme
- Configuring the authorization scheme in the application
- Defining the product operational structure
- Defining the operator attributes

Supported authentication schemes

Starting with Pega 7.3.1, the following authentication types are supported:

- PRBasic - Based on passwords in the Operator ID data instances and the login-in form (defined by the HTML rule @baseclass.Web-Login, which your application can override).
- PRSecuredBasic - Similar to PRBasic but passes credentials using Secure Sockets Layer (SSL) using Basic HTTP authentication. The login form is defined by the

HTML rule @baseclass. Web- Login-SecuredBasic, 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 Pega 7.3.1 is deployed using JAAS to authenticate users.

Defining an authentication scheme

Pega supports the use of a centralized, automated means of maintaining operator data instead of maintaining it in the Product Designer for Financial Services application.

Use the **Security model** portion of the implementation planning workbook for the Pega application being implemented to record your decisions during this procedure.

1. Discuss Authentication schemes with your site's security and application server teams.
2. Determine the appropriate authentication model for the organization by defining the access group, portal, and roles for each user type.

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

Product Designer for Financial Services includes the following pre-defined operators and access groups. You can log in to the application using one of these Operator IDs using the password that you specified when you enabled this operator; to enable Pega-provided operators, see [Enabling operators](#).

Operator ID	Access Group	Role
PDFSDesigner	PegaPDFS:Designer	PegaPDFS:ProductDesigner
PDFSMarketer	PegaPDFS:Marketer	PegaPDFS:ProductMarketer
PDFSManager	PegaPDFS:Manager	PegaPDFS:ProductManager

Operator ID	Access Group	Role
PDFSSysAdmin	PegaPDFS:Administrator	PegaPDFS:Administrator
		PegaPDFS:ProductManager
		PegaPDFS:ProductDesigner
		PegaPDFS:ProductMarketer
PDFSPowerUser	PegaPDFS:PowerUser	PegaPDFS:Administrator
		PegaPDFS:ProductManager
		PegaPDFS:ProductDesigner
		PegaPDFS:ProductMarketer

Use these access groups, roles, and privileges for your site needs to control access to the functionalities in the application implementation. To configure an authorization scheme in the application, do the following:

1. Review the Product Designer for Financial Services access groups and roles:
 - To view access groups: Click **Dev Studio > Org & Security > Groups & Roles > Access Groups**. The access groups use the preface PegaPDFS.
 - To view roles: Click **Dev Studio > Org & Security > Groups & Roles > Roles**. The roles use the preface PegaPDFS.
2. Define access groups for the application. The following access groups are pre-defined in the application:
 - PegaPDFS:Administrator
 - PegaPDFS:Designer
 - PegaPDFS:Manager
 - PegaPDFS:Marketer
 - PegaPDFS:PowerUser

3. Record the authentication scheme in the Security model worksheet in *your project's Implementation Planning Workbook*
4. Identify additional access groups needed for your application.
5. Identify portals associated with these access groups.
 - Define the access roles and privileges for the application.

Pega authentication schemes work by associating one or more roles to 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. The application comes with the following pre-defined roles:

- PegaPDFS:Administrator
- PegaPDFS:ProductDesigner
- PegaPDFS:ProductManager
- PegaPDFS:ProductMarketer

For more information, see [Access group and role configuration](#).

Defining the product operational structure

Leverage the product operational structure for routing and reporting within the application. Typically, the product operational structure does not map operators exactly to your products organization but instead, it maps the work that those operators do.

To support complex product structures, the standard Pega Organization and Security functionality is extended in the application. Use this functionality to create and name levels in the product hierarchy between the Financial Institution level and other lower levels such as Business Line and Product Category.

Product Designer for Financial Services includes a sample product structure and its associated taxonomy. You should familiarize yourself with the intricacies of this organizational structure and its use by the set of Demo operators provided in the PDFS Demo application.

Use the Organization structure worksheet in the *Implementation Planning Workbook* to record your decisions during this procedure.

1. Log in to the application as **PDFSSysAdmin** using the password that you specified when you enabled this operator; to enable Pega-provided operators, see [Enabling operators](#).
2. Click **Dev Studio > Financial Services > Product Structure**. The sample product organizational structure is named “Uplus FS Product Operations”.
3. Review the existing structure and taxonomy.
4. 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, these fields in the operator record influence how the application handles assignment of work to the user:

- Work group
- Calendar

For most implementations, it is a best practice to configure your application to set values on the operator record during the authentication process rather than requiring an administrator to manually maintain these records. You must configure these rules as part of the authentication mechanism for your site. For more information, see the Pega 8.4 help topic [More about Authentication Services](#).

Defining the operator work group

The work group setting on the operator record affects how the application delivers work to the operator. Use the Product structure worksheet in the *Implementation Planning Workbook* to record your decisions during this procedure. Do the following for each operator:

1. Click Dev Studio > Org & Security > Organization > Operators.
2. Click Operator ID.

3. On the **Work** tab for this **Operator ID**, review the work group information for the operator record.
4. Determine your policy for assigning a work group to an operator or role that multiple operators hold.
5. Configure the operator with the role that is appropriate to your organization.
6. You can assign multiple operators to the same role.
7. Save the configuration.

Defining the operator calendar

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.

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. Use the Organization structure worksheet in the *Implementation Planning Workbook* to record your decisions during this procedure.

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

For more information, see [More about Calendar data instances](#).

To thoroughly learn about the Pega security model, enroll in the *Lead System Architect* course on Pega Academy.

Defining the product components

Product Designer for Financial Services provides a **Library** page on the **Product Studio** portal to help you create the components needed to create products and bundles.

Components are defined in the library and are available to be reused when creating products and bundles. Components are created and maintained with actions that are

accessible via the user's role. All users can view a component or view its references. Only users with the appropriate roles can create, configure, retire, and delete components.

- Create
- View
- Configure – Update the configuration of a component.
- Retire – Not available for use in products but can continue to be referenced by products that already use them. Retired components appear in the library.
- Delete – Not available for use in products and are not used in any existing products. Deleted components do not appear in the library.
- View references – View list of products or components that use the component.

Each component has a name, unique identifier and description as well as other fields that are relevant to the component. The unique identifier is set to the component name but can be updated to a different value if desired. Components have a status of 'Active' in the library when they are available for use in a product or bundle.

The Product Designer and Product Marketer creates and manages the following components.

Attributes

Attributes are the data elements of a product. Users with the Product Designer and Product Marketer roles can create and maintain attributes. An attribute consists of:

- Attribute Name
- Attribute Description
- Attribute type
 - Product: Identifies characteristics about the product.
 - Adjudication: Data elements captured during onboarding and used during the adjudication process.
 - Legal: Legally required data elements captured during onboarding.

- Core banking system: Data elements required by the core banking systems during fulfillment.
- Data type
 - Text (single line)
 - Text (paragraph)
 - Boolean
 - Currency
 - Date & time
 - Date only
 - Decimal
 - Email
 - Integer
 - Phone
 - Picklist
 - Time only
 - URL

Eligibility rules

Eligibility rules define the criteria, which your customers must meet for a product to be offered or sold to them. Users with the Product Marketer role can create and maintain each Eligibility rule, which consists of:

- Eligibility name
- Eligibility description
- Eligibility details – A full text field that describes the criteria of the eligibility in enough detail that the application using the product can implement the criteria logic.

Benefits

A benefit describes a distinct value the product brings to the product. Users with the Product Marketer role can create and maintain Benefits, which consists of:

- Benefit name
- Benefit description
- Benefit type
 - Account – Identify the account name
 - Service - Identify the service name
 - Other – A benefit value that is neither an account or service

The Product Designer for Financial Services application features the ability to automatically provide a bulleted list of benefits for a product: when you associate benefits with a product, the application concatenates the benefit descriptions into a string and inserts this into the BenefitsList column of the Product Matrix table.

Prices

A price rule defines a single pricing element of a product. Users with the Product Marketer role can create and maintain Price rules. A price rule supports a number of pricing configurations, which consists of:

- Price name
- Price description
- Category – Specifies the node in the product structure hierarchy that defines the scope for the products that can include this price. For example, a price rule at the **Retail > Lending** node in the product structure hierarchy, **Retail > Lending > Secured > Mortgage**, is available for use by all products at the **Lending**, **Secured**, and **Mortgage** nodes.
- Currency – Specifies the currency of price amount. Leave this value blank to indicate that the application uses the currency you defined for pricing in your organization's operational structure.
- Type
 - Fee – An amount charged for a service. You can configure the following Fee settings: - Fee occurrence
- Once – A one-time charge

- Recurring – A recurring charge with a frequency of annually, monthly, quarterly or semi-annually
- With account activity – A charged based on certain activities on the account. You can add the following details to an account activity:
- Account activity type – You can configure the following types of account activity:
 - Cash advance
 - International travel
 - Missed payment
 - Overdraw
 - Other
 - Description of fee circumstances – An optional text fields in which you can add information around the circumstances of an account activity.
 - Value – The value of the price. You can add the following details for the value:
 - Fee amount – Sets an amount for the price
 - Percentage of attribute – Sets a percentage by which the application calculates a price as a percentage of a product attribute. You configure which product attribute from the list of attributes in the library on which the application calculates the price value.
 - Value configuration – Specifies the price value. You can add the following details for the price value:
 - External document – Specifies that the price value is sourced from a file that can be uploaded to the price rule or a URL of an external document
 - Lookup table – Specifies that the price value is sourced from a lookup table you can configure in Product Designer for Financial Services
 - Type value – Specifies a hard-coded value
 - Set range – Defines the minimum and maximum values that the amount must fall within upon modification when the product is offered to a customer - Rate – An amount charged for the use of the product.

You can configure the following Rate settings:

- Rate type – Fixed. You can add the following details for a fixed rate: -
 - Frequency – Specifies the frequency of when the rate is charged – annually, monthly, quarterly, and semi-annually
- Rate base attribute – Specifies a product attribute from which the application calculates a rate
- Value configuration - Specifies the source for the rate value. You can add the following details for a fixed rate value:
 - External document – Specifies that the rate value is sourced from a file that can be uploaded to the price rule or a URL of an external document
 - Industry benchmark – Specifies that the application calculates the rate value as a specified percentage over the LIBOR or Prime rates
 - Lookup table – Specifies that the rate value is sourced from a lookup table defined in Product Designer for Financial Services
 - Type value – Specifies a hard-coded rate value type
 - Set range – Defines the minimum and maximum values that the amount must fall within upon modification when the product is offered to a customer.
- Rate type – Variable
 - Term intervals – A series of time intervals that defines how the specified variable rates are applied over the life of the account
 - Interval length – Length of the interval –
 - Value configuration - Specifies the source for the variable rate value. You can add the following details for the rate value:
 - External document – Specifies that the rate value is sourced from a file that can be uploaded to the price rule or a URL of an external document
 - Industry benchmark – Specifies that the application calculates the rate value as a specified percentage over the LIBOR or Prime rates –
 - Lookup table – Specifies that the rate value is sourced from a lookup table defined in Product Designer for Financial Services
 - Type value – Specifies a hard-coded rate value type

- Rate floor – Defines the value that the rate cannot go below during the life of the account
- Rate ceiling – Defines the value that the rate cannot go above during the life of the account

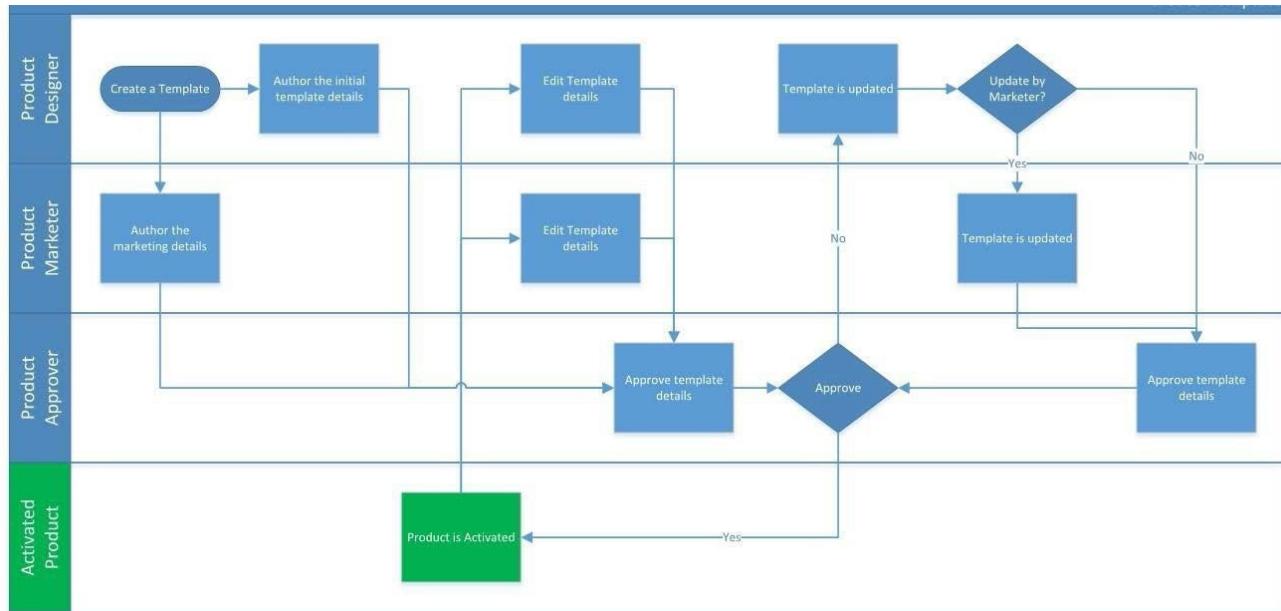
Templates

You can create products from templates, which define the characteristics of a set of similar products. A template is configured such that you can quickly create a product using all the attributes, eligibilities, benefits, and price rules that make up each of these products. Templates are library components and products are in the Product Catalog to be offered and sold to a customer. For example, a fixed mortgage template is defined to have all the characteristics of a fixed mortgage, and you can use this one template to create 15-year, 20-year, and 30-year mortgage products.

You configure template components to be required and/or read-only: *required* components cannot be deleted from the products created from the template, while *read-only* components cannot be updated in the products created from the template. These configuration elements in a template are called **lock options**.

You manage a template using a Template case. Product Designers and Product Marketers collaborate to build their product templates. A user with the *Product Designer* role can create templates and configure attributes in the template. A user with the *Product Marketer* role can configure the Prices, Benefits, and Eligibilities for a template. Any user with both Designer and Marketer roles can perform both sets of actions on a template. When completed, the template is routed to a Product Manager for final approval before it is available for use to create products.

The following chart illustrates the template lifecycle flow:



Building features

Follow these steps to build application features:

Updating your application settings

Product Designer for Financial Services includes system settings that identify the class and rulesets used to implement the elements of the Product Catalog. To view or change these settings, click **Dev Studio > Financial Services > PDFS Configurations** to go to the **PDFS Configurations** landing page in the Dev Studio.

The three system settings are:

- **Product Catalog Ruleset** - Holds the library elements, products, and bundles in the product catalog. This value is defaulted to **PegaPDFSCatalog**. It is intended for you to use this ruleset for your Product Catalog. You should only change it under two circumstances:
 - If you want to use the sample product catalog provided with the Demo application.

- For details on making this change, see the section for installing the Demo application in the *Product Designer for Financial Services Installation Guide* available on the Pega Foundation for Financial Services [product page](#).
- If you choose to use custom rulesets instead of the Pega-provided ones.
- **Product Data Class** - Holds the root class for the data elements of product catalog. You should not need to modify the Pega-provided default value.
- **Product Library Class** - Holds the root class for the data elements in the library. You should not need to modify the Pega-provided default value.

Creating library components

Using the Product Studio portal, create the following library components:

- Attributes
- Prices
- Benefits
- Eligibilities
- Templates

After you determine the library assets that are needed to create your products, go to the appropriate tab on the Library page in the Product Studio portal.

You should create the Attributes before creating the Price, Benefit, and Eligibility assets. All of these assets need to be created in the library before you can include them in the template.

Creating products and product bundles in the Product Catalog

Product Designer for Financial Services users with the Product Designer role can create products and bundles. You create bundles to offer multiple products and services as one combined product. A product bundle is associated with a single market segment but can consist of products from different Lines of Business and Categories.

To create products, complete the following steps:

1. In the Product Studio, click the tabs: **Catalog > Products**.
2. On the Products screen, click **Add product**. The Add product screen appears.
3. Enter **Name** and **Description** for the product you want to create. Modify the identifier if desired. Choose the **Type** and **Category** values. Select the template from the available templates in the **Use template** field. Enter the **Active from** and **Expires on** date fields.
4. Click **Submit** to create the product.

The product is now ready to be worked on by the Product Designer and Product Marketer. The product remains in the Author stage with a status of **Draft** while you configure it.

5. Click **Edit** to make changes or configure the attributes, prices, benefits, and eligibilities that were defaulted from the template. Click **Done editing** when you are finished with your changes. Product Designer for Financial Services allows you to work on a product with different editing sessions over time.
6. Click **I'm done** when you complete configuring the parts of the product on which you are working. When both the Product Designer and Product Marketer have clicked **I'm done**, the product is routed to the approving manager for approval.
7. To approve the product, the manager opens the product from the approval work queue, enters comments, and clicks **Accept** or **Reject**. Accepted products are automatically published in the Product Catalog, which makes it available to offer and sell to your customers. Rejected products are automatically routed back to the Product Designer to be updated appropriately.

To create and configure a product bundle, complete the following steps:

1. In the Product Studio, click the tabs: **Catalog > Bundles**.
2. On the Bundles screen, click **Add bundle**. The Add bundle screen appears.
3. Enter the **Name**, **Identifier**, **Description**, and the Active from and expires on dates for the bundle you want to create.
4. Click **Add new** to add products to the bundle. You can add products across various Lines of Business and Categories.
5. After adding products to a bundle, set the **option** for the products in the bundle.

6. Click **Submit** to create the bundle. The bundle work area displays.
7. Click **Edit** on the header to add details to the bundle.
8. To add or delete products in the bundle:
 - Click **Add new** in the products section to add products to the bundle.
 - Click on the trash can delete icon to remove the products from the bundle.
9. By default, the application copies eligibility rules that are associated with the products into the bundle; you can see these eligibility rules in the bundle by selecting the **Eligibilities** tab in the bundle work area; however, you can add additional and delete undesired eligibility rules from the bundle in this screen. When you are satisfied with the eligibility rules associated to the bundle, continue to the next step.
10. Click on the **Benefits** tab to add additional and delete undesired benefits in the bundle. When you are satisfied with the benefits associated to the bundle, continue to the next step.
11. The products in the bundle are shown as expandable/collapsible sections in the **Product modification** section. Go to the **Product modification** section to modify the prices and benefits that were defined on the products:

Changes here do not affect the pricing and benefits at the individual product level; they are saved only in the bundle specific pricing.

- To modify a price: Under the **Pricing** tab, you can modify the prices that were defined in the product. Click **Modify price**. Click on any price row you wish to modify.
 - The price details are shown in a modal. To waive the price for this bundle, select **waive** and click **Submit**. Waived prices are identified with a status of "waived." - To edit the price for this bundle, modify the rate or percentage values and click **Submit**. Edited prices are identified with a status of "edited." - To revert any of your changes, click **undo**.
 - To see the price details, click on "gear" icon.
- To modify or add a new benefit: The expandable/collapsible sections in the **Product modification** section allow you to modify bundle benefits. Under the **Benefits** tab, click **Modify benefits**.

- The benefit details are shown in a modal. To waive the benefit for this bundle select Waive and click Submit.
- To edit the benefit details for this bundle, update the details appropriately and click Submit. The price benefit are shown with the status “edited.” –
- To revert any of your changes, click undo.
- To add new benefits for a product that is as part of the bundle, click Add new, which displays all the benefits in the library. The benefits which are currently part of the product have a green checkmark. Click + to add benefits to the product as part of the bundle.
- After you add a benefit, you can also modify the new benefit details as appropriate to bundle requirements.
- The newly added benefits are displayed with Active status. You can also delete them by clicking the trashcan icon.

After adding these details to the bundle, you must send this bundle for approval. The approval process of the bundle is similar to the product approval process described above.

Production maintenance and monitoring

Production maintenance and monitoring includes:

Business rule maintenance in the production environment

You can give managers the ability to update other rule types in the Production environment. For example, managers can update the Goals and Deadline for a certain case type. These rules must be delegated in Dev Studio first. After the rule has been delegated, you can access it by clicking your operator profile > My rules. For more information on rule delegation, see [Delegating a rule or data type](#).

Application health monitoring

Autonomic Events Services (AES) is an application that automatically monitors, retrieves, and organizes the alert data from one or more clustered systems throughout the enterprise.

Pegasystems also provides the [Predictive Diagnostic Cloud](#) (PDC) which allows you to benefit from AES without installing it locally. PDC is a Software as a Service offering of AES.

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

Identifying and reporting issues

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

- Identify operational staff who are responsible for responding to issues reported in the production environment.
- Establish procedures with these resources to triage, respond to, and escalate issues.
- Determine procedures for delivery of changes to the production environment.

Technotes

- [Pega Foundation for Financial Services Accounting Guide](#)
- [Pega Foundation for Financial Services requirements guide](#)
- [Pega Foundation for Financial Services requirements portal guide](#)
- [Pega Foundation for Financial Services Version Verification Tech Note](#)

Pega Foundation for Financial Services Accounting Guide

- [About this document](#)
- [Accounts](#)
- [Payment types](#)
- [Accounting verification](#)
- [Accounting cutoff](#)
- [Accounting rules](#)
- [Using automatic processing flows and flow actions](#)
- [Duplicate and offset processing](#)
- [Verifying accounting functionality](#)

About this document

This document describes the accounting elements of the PegaAccounting and PegaAccounting- Classes rulesets that collectively make up the accounting components of the Pega® Foundation for Financial Services application.

The topics covered in this document include:

- Accounts
- Payment types
- Accounting verification
- Accounting cutoff
- Accounting rules
- Automatic processing flows and flow actions
- Duplicate and offset processing
- **Intended audience**

Intended audience

This guide is intended for developers who are customizing these rules during initial product design and deployment.

Accounts

- **Account constructs**
- **Account setup and operation**
- **Internal accounts**

Account constructs

A Pega Accounting account is a unique structure combining four separate and required attributes.

- **Account Type (AccountType)** – A four-character alphanumeric value such as GL, DDA, or 3725 that precedes the account number. It categorizes the account, and allows differentiation for display, accounting entries, and reconciliation purposes. In addition, types serve as the data points for the sorting and posting of different types of accounts in the system's accounting feed file.

- **Account Number (AccountNumber)** – A numeric value that follows the account type and defines the account number to which the debit or credit entry is posted.
- **Bank Number (BankNumber)** – A four-character alphanumeric value such as 0001 or CA02 that identifies the bank associated with the account number processed by the accounting entry. By requiring a bank number with every financial transaction, Pega supports multi-bank and multi-organizational accounting needs.
- **Account Name (Name)** – An alphanumeric value that identifies the account's owner name.

Account setup and operation

You should configure an accounting step detailed in the Configuring the Accounting Step Rule section of this guide to pass the account information to accounting transactions. Depending on the information you want to pass, it may require the setup of one or more of the following components.

- Internal accounts
- Chart of Accounts
- Accounting party data

Internal accounts

Internal accounts let you use symbolic names for accounts that are repeatedly used in accounting transactions and are typically not a party in the case. Internal accounts are commonly used to define suspense and write-off accounts, track paper or dummy entries, departments or external processes originating exceptions, and consolidated master bank accounts where one internal account number is used regardless of which member in a specific banking group is being debited or credited (for example, the Federal Reserve).

Internal accounts form the basis for a look up to a table called the Chart of Accounts. Internal accounts are instances of PegaAcct-Party-InternalAccount that you can copy into your application's ruleset.

The following is an example of an internal account instance for a suspense receivable (debit) account.

- **Chart of Accounts**
- **Chart of Accounts lookup – how it works**
- **Accounting party data**

Chart of Accounts

The chart of accounts associates internal accounts with transactions in an accounting rule. When a debit or credit is defined in an Accounting Step rule to default from the Chart of Accounts, the system performs a lookup of this table to get the account type, account number, bank number, and name of the internal account, and maps it to the debit or credit transaction.

To add or modify, save a copy of the Chart of Accounts – a map value rule named FinChartOfAccts to your ruleset and update it with references to your internal accounts. The following screen shot shows the Chart of Accounts.

Account	Default	Bank Number
Payable	Payable	Payable
Receivable	Receivable	Payable
WriteOff	WriteOff	Payable
Offset	Offset	Payable
Fee	Fee	Payable
ChargeFee	Fee	Payable
Default	ERROR	Payable

Chart of Accounts lookup – how it works

The lookup is executed in the FinChartOfAcctsLookup activity in the Embed-Rule-PegaAcct- Financial-Txn class.

The activity first uses the FinWorkPoolClasses map value rule using the class of the work object and an input value of InternalAccount to return the name of the class holding internal accounts (by default, the class is PegaAcct-Party-InternalAccount). The following screen shot shows the FinWorkPoolClasses map value rule.

Class Type	Default
InternalAccount	PegaAcct-Party
Default	PegaAcct-Party

This map value rule finds the correct classes to use for a work pool.

The activity then uses the Property-Map-ValuePair method rule to find the actual key for the InternalAccount. To do this, it uses the FinChartOfAccts map value rule using the class of the work object and input values of the symbolic (from the property AccountFromValue specified in the step rule) and the value of the work object property .InternalBank, to look up accounts based on BankNumber.

You can set the .InternalBank property using standard methods such as Property- Set, Data Transform, or Rule-Declare-Expressions, to the BankNumber associated with the case's account.

Accounting party data

In addition to getting account information from an internal account, accounting steps also default account information from WorkParty information that is entered or defaulted into a work object.

When the StepRules reference a WorkParty role and the accounting is performed in the background or the party information is defined in the rule so that you cannot modify

the account data, the specified role in the work object must contain data in the following properties. If you can modify account information, you can enter this data online.

The following properties are defined in the PegaAcct-Party- class.

- .AccountNumber
- .AccountType
- .BankNumber
- .Name

Typically, during deployment, you extend these classes. When a new Party class is defined, it must be derived from this class. There is no need to define new properties; ensure that values are set when you perform the lookup of the WorkParty information.

Payment types

Payment types define two key decision variables in an accounting step related to the processing of debit and credit entries.

- Whether accounting advices, notifications, or documents are generated with the debit or credit entry, and in which format (e-mail, mail, and so on).
- The accounting cutoff time for the transaction.

These are instances of Rule-PegaAcct-Financial-PaymentType and usually apply to the PegaAcct- Txn class (the specific Accounting class).

- **Referenced by**
- **Format**

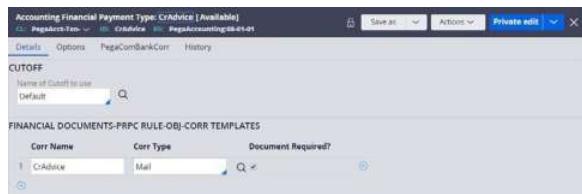
Referenced by

Payment types are referenced by the Pymt Type field of a transaction in an accounting step rule.

Format

The Payment Type name can be mixed-case alphabetic characters with no spaces or special characters.

Note: Adopt a naming convention that identifies the type of accounting transaction that the payment type applies to. For example, you can create GLNONE for debit and credit suspense transactions that are GL accounts and do not generate accounting documents. After creating, you can apply this payment type universally to all accounting step transactions referencing suspense.



The following screen shot shows the payment type instance Rule-PegaAcct-Financial-PaymentType.CrAdvice.

Field	Description
Short Description	Required. Short description to describe the Payment Type.
Cutoff Section	Cutoff Section
Name of Cutoff to Use	Name of the cutoff.
Financial Documents Section	
Stream Name	The name of the document to generate for the accounting entry. Specify a correspondence rule instance.

Corr Type	The type of correspondence or document – mail, email, fax, and so on from the prompt list.
Required?	Check if this document is required when the accounting entry is generated.

Accounting verification

The accounting component supports multiple levels of accounting verification. These levels are based on the total value of the debit or credit in the accounting transaction. The verification level also represents the number of times a transaction must be verified before it can be posted to the system's accounting feed file.

Privileges assigned to an operator determine the dollar levels that operators have the authority to verify. Although an operator can have the security to verify at multiple levels, an operator cannot verify a single transaction more than once.

The recommended approach is not to require verification for accounting steps processed automatically at entry, including small dollar write-off and financial offset processing.

Because these transactions are workflow-based, and the operator cannot intervene. This approach avoids unnecessary processing interruptions of the workflow and enables a higher level of straight-through processing.



Note: None of the accounting steps installed with Pega Foundation for Financial Services has verification enabled.

Verification levels (and flow names) are set in the **FinVerificationFlowByAmount** map value rule. In the accounting rules, you specify an activity called **FinVerifyByAmount** in

the **Activity for Verification Flow** field on the Step tab of the rule. This activity calls the mapped value during processing to set the verification level or flow of the transaction.

- **Referenced by**
- **Format**
- **Accounting records affected**

Referenced by

The accounting verification process is referenced in the following.

- Accounting steps
- Verification flows and workqueues

Format

The threshold amount is a numeric – whole number or 2 decimals without commas.

The amount entered is evaluated as an amount that is up to and including that amount.

Accounting records affected

The FinVerificationFlowByAmount map value rule, shown in the following screen shot, defines verification levels.

Amount Thresholds	Default
< 100	FinStepVerifNot
< 1000	FinStepVerifOr
< 5000	FinStepVerifTwo
Default	FinStepVerifThr

For deployment, you can build verification to evaluate other levels by creating a flow and referencing it in the map value rule. It comes with three levels of verification already set up for you. Complete the following steps.

- Review the verification values.

- Copy the flow and map value rule to your ruleset.
- Update to meet your institution's requirements.
- Update the accounting rules for which you want to invoke the verification process.



Note: Offset processing sends both cases to verification at the same time to maintain balanced processing.

Accounting cutoff

The accounting component supports the use of warning and cutoff times for accounting transactions. This enables your organization to put a logical stop time to accounting and then run end-of-day accounting feeds and reconciliation and balance reports.

You can use multiple cutoff rules to configure different accounting extraction and feed schedules. You can activate cutoff in two modes:

- Standard – Allows you to continue researching and processing cases but prevents accounting transactions from posting until a specified release time is reached. In most instances, work on the case can continue until a workflow or transaction point is reached where processing cannot advance until the transaction is posted. At installation, this is the default cutoff mode.
- Future – Allows you to continue researching and processing cases and continue posting transactions after the day's cutoff has been reached. When processing in this cutoff mode, debit and credit transactions in the step are posted to the accounting posting file with the next business day's date as the posting date. This means that the accounting feed must be extracted by posting date, which is the date the transactions should post, not the posting timestamp, which is when the transaction was written to the accounting posting file. The online display shows both the date and time written to the posting file and the date specified for actual posting, so you know what was done between the cutoff and the next business day. This method also works well for weekend processing.



Note: Transactions held for cutoff are always held as balanced transactions. Debits and credits are not separated prior to posting.

- **Referenced by**
- **Format**
- **Accounting records affected**
- **Cutoff operation**

Referenced by

- The accounting cutoff process is referenced by:
- Payment types used in the accounting step rules
- Accounting displays that identify entries held for cutoff
- Online indicators that identify entries that passed warning or cutoff times

Format

- Time – Set in 24-hour notation format of hhmmss.
- Naming scheme – Describes the accounting deadline (that is, DDA or General Ledger).

Accounting records affected

- **Map value rule for cutoff type** – Defines the type of cutoff, defaults to hold for standard mode, and update to future for future mode.
- **Map value rule for cutoff calendar** – Defines which calendar to reference for work dates.
- **Rule-PegaAcct-Financial-Cutoff instance** – Defines the cutoff time, one instance for each cutoff schedule needed.



Note: If you do not use cutoff functionality during initial deployment, you do not need to copy the records.

The following screen shot shows the cutoff type configuration:

Column Name	
Cutoff Type	Default

Default Hold

The following screen shot shows an example of the cutoff calendar form used to select which cutoff calendar to use:

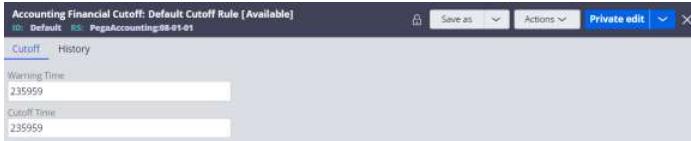
Day	Business	Start	End
Sunday	Business	Start	End
Monday	Business	Start 010000	End 235959
Tuesday	Business	Start 010000	End 235959
Wednesday	Business	Start 010000	End 235959
Thursday	Business	Start 010000	End 235959
Friday	Business	Start 010000	End 235959
Saturday	Business	Start	End

Closed days
1 Jan 1, 2017

The following screen shot shows an example of the cutoff calendar rule used to define the cutoff times.

Default	Value
	AccountingDefs

The following screen shot shows the Financial Cutoff rule.



Cutoff operation

The cutoff time in the transaction and step class is a string property and is in the format of HHmmss. It does not include a reference to time zone. During processing, the time is compared with the current time and time zone of the server processing the transaction to determine if the cutoff time has been reached.

For example, if the server is in CMT, set the cutoff times to the value in CMT. If the server is in CMT and the accounting deals with a processing area in EST whose cutoff is 18:00 EST, then set the cutoff to 17:00 (the CMT value). The 17:00 time is then compared with the time portion of the current date on the server (which is in the server time zone).



Note: Cutoff times (warn and cutoff) are stored in each transaction and the earliest transaction time is copied to the Step.

The rules that compare current time to cutoff times are:

- Function rule named PegaAcct Default PastStepCutoff – Compares the hour and minutes of the current time (using the server's time zone) with the step cutoff.
- HTML fragment rule named FinTxnCutoff – Compares the hour and minutes of the current time (using the server's time zone) with the transaction cutoff warning or cutoff times. This is used to display the red or yellow clocks on the worklist if a transaction is past warning or cutoff.

Checking to determine if a step is past cutoff occurs in the FinStepPost flow in the work object class using the activities FinStepCutoffEval in the work and step classes. The processing of accounting which is past cutoff differs based on a map value.

- Hold cutoff accounting
- Future cutoff accounting

Hold cutoff accounting

When the PegaAcct-Step-FinCutoffType map value rule is set to Hold, the following cutoff processing occurs:

- If the step is past cutoff, then an assignment is created in a cutoff work queue to wait for release on the next business day in the calendar.
- Two activities set the release time (goal time) of the SLA for accounting held in cutoff. Both set the release time to the GMT value (using Pega date functions) from the date calculated via Java GregorianCalendar functions (which are local time zones).
- The **FinWorkBasketAccounting** activity in the Work- class sets the goal time for the assignment SLA which waits before testing if accounting can be released. It sets the goal time based on the hours and minutes, which it gets from the **FinReleaseTime** data transform in the Rule-PegaAcct-Financial-Cutoff class, and uses java GregorianCalendar to add a day. It uses the hour attribute of the time (which again is local time zone). So, if the local time is between 00:00 and 00:59 it does not add a day, it just sets the hours and minutes – so the release time is later that day. The assignment pxGoalTime is set to a GMT value and correctly adjusts the date's time using the Pega java routine DateTimeUtils fomatDateTimeStamp (java.util.Date).
- The **FinCheckCutoff** activity in the Work- class resets the goal time when the SLA starts and is still past cutoff (has to be held). It just adds one day to the time using the GregorianCalendar in the same way as the activity above, which sets the original SLA time. It does not contain the logic for between midnight and 1 A.M.

Future cutoff accounting

When the PegaAcct-Step-FinCutoffType map value rule is set to Future, the following processing occurs:

If the step is past cutoff, then instead of creating an assignment to wait, the step is posted but with the next business day. The cutoff evaluation in the FinStepPost flow sets the task status to ready for release.

If the step is not past cutoff, the evaluation activity changes the value of the cutoff type, so it is no longer Future.

During posting, the **PegaAcct-Step-FinPostStep** activity is used to set the posting date and time. If the CutoffType is Future, it uses the Rule-Utility-Function BusinessCalendar.NextBusinessDay. If the CutoffType is no longer Future, this activity sets the posting date and time in GMT.

Accounting rules

Accounting steps are either an adjustment or a suspense transaction.

Accounting Step rules define the debit and credit transactions that you should process as a **contained** bucket of work. Each step can contain 1 to N debits and 1 to N credits. The only requirement is that the sum total of the credits must equal the sum total of the debits at run time. At design time, there is no validation on the amount.

- **Adjustments**
- **Suspense**
- **Outstanding balance**
- **Suspense actions – Open, Raise, Lower, Close, and WriteOff**
- **Configuring the accounting step rule**

Adjustments

Adjustment rules define a one-time accounting transaction that creates a balanced set of debit and credit account entries. This is a one-time transaction that has no life beyond the transactions.

Examples include debiting a customer and crediting a fee account as if you are collecting a fee for a service rendered or oppositely crediting a fee back to a customer as a fee reversal. Adjustment actions that can be associated with an accounting step are either Adjust or WriteOff.

Suspense

Suspense rules define a series of accounting transactions that create a balanced set of debit and credit entries, one of which is to a suspense account. Suspense transactions have a life beyond the initial transaction. They live on, and detailed sub-ledger balances are maintained until the suspense item is resolved and closed.

There are two basic types of suspense categories: Payables and Receivables.

- **Payables**
- **Receivables**

Payables

Payables are opened as credits in suspense ledgers, with the presumption that monies are due to another party at some future point. For example, responding to a customer's complaint that a credit to their account is unrecognizable or that a credit, coming in from some ACH transaction, cannot find an appropriate home, and so on. In each case, a debit is processed to satisfy the customer or offset the ACH transaction and a credit is set up waiting for final disposition.

Receivables

Receivables are opened as debits in suspense ledgers, with the presumption that monies are due from another party. For example, responding to customers' complaints that they do not recognize a debit to their account or that a debit, coming in from an ACH, cannot be posted for some reason and so on. In each case, a credit is processed to satisfy the customer or offset the ACH transaction and a debit is set up waiting for final disposition.

Outstanding balance

An outstanding balance is calculated each time an action is executed against the original suspense entry. The balance is first created and set when you open the suspense entry. The balance is tracked and adjusted as you take subsequent accounting actions and steps on the balance. Work Objects and Cases cannot be resolved until the outstanding balance reaches 0.00.

Suspense actions – Open, Raise, Lower, Close, and WriteOff

Suspense actions that can be associated with a suspense step are:

- **Open** – Creates a new suspense entry
 - Receivable – Adds a debit to the receivable ledger (debits receivable, credits x)
 - Payable – Adds a credit to the payable ledger (credits payable, debits x)
- **Raise** – Increases the value of the suspense
 - Receivable – Increases the value of the debit in the receivable ledger and outstanding balance (debits receivables, creditsx)
 - Payable – Increases the value of the credit in the payable ledger and outstanding balance (credits payables, debitsx)
- **Lower** – Decreases the value of the suspense
 - Receivable – Decreases the value of the debit in the receivable ledger and outstanding balance (credits receivables, debits x)
 - Payable – Decreases the value of the credit in the payable ledger and outstanding balance (debits payables, creditsx)
- **Close** – Resolves the suspense – Lowers the outstanding value to zero
 - Receivable – Decreases the value of the debit in the receivable ledger by the total value of the outstanding balance (credits receivables, debits x)
 - Payable – Decreases the value of the credit in the payable ledger by the total value of the outstanding balance (debits payables, credits x)

- **WriteOff** – Resolves the suspense as a write-off – lowers the outstanding value to zero
 - Receivable – Decreases the value of the debit in the receivable ledger by the total value of the outstanding balance (credits receivables, debits a WriteOff account as defined in the chart of accounts)
 - Payable – Decreases the value of the credit it in the payable ledger by the total value of the outstanding balance (debits payables, credits WriteOff account as defined in the chart of accounts)

Configuring the accounting step rule

Accounting step rules share the same class structure, including the key. The key consists of the class of the case for which the accounting is being processed, the accounting action, and a step descriptor or type.

When deploying accounting functions, you need to analyze and update a number of data elements in the Accounting Step rules. To modify a step rule, first save it in your ruleset; then update it. The updates typically include one or all of the following tasks:

- Adding verification to a step.
 - Updating the payment type.
 - Changing default account information to use different party roles and amount properties.
 - Referencing different internal accounts listed in your Chart of Accounts.
- **Rules affected**
 - **Accounting step rule fields**
 - **Naming conventions for accounting step rules**

Rules affected

The accounting step rules provided are instances of three classes:

- **Rule-PegaAcct-Financial-Adjustment** – for adjustments

- **Rule-PegaAcct-Financial-Payable** – for suspense payable (credit) entries
- **Rule-PegaAcct-Financial-Receivable** – for suspense receivable (debit) entries

The following shows the **Step** tab of a Receivable rule for the Rule-PegaAcct- Financial-Receivable.CrAcctOwner instance.

This screenshot shows the 'Step' tab of a Pega application interface. The title bar indicates it's for an Accounting Financial Receivable rule. The main area contains several input fields: 'Class of the Step' set to 'PegaAcct-Step-Open', 'Model to create Step' set to 'Activity for Verification Flow', and a 'Likelihood' slider set to 5. Below these are three dropdown menus for 'Value Dates': 'Balance', 'Different', and 'In Past'. There are also some small checkboxes or buttons labeled 'Save as', 'Actions', and 'Private edit'.

The following shows the **Transactions** tab for the same instance; the screen is wide, and you must scroll from left to right to see all the fields. These fields, and their use in creating new step rules, are described in detail below.

This screenshot shows the 'Transactions' tab of the same application. It displays a very long list of transaction-related fields, likely over 20 items, arranged in a grid. The fields include various identifiers like 'Step', 'Line', 'Ref ID', 'Ref Type', 'Ref Account', 'Initial Amount Due', 'Initial Due Date', 'Initial Due Month', 'Initial Due Year', 'Initial Due Day', 'Initial Due Hour', 'Initial Due Minute', 'Initial Due Second', and 'Initial Due Millisecond'. Each field has a dropdown arrow indicating it's a selection field.

Accounting step rule fields

Field	Description
Short Description	Required. Short description of the rule. This value is displayed during manual accounting in the list of step rules from which you can select.
Step tab	
Class of the Step	The class of the page that holds the step data. Specify a Rule-Obj-class name that is a concrete class that is a child of class PegaAcct- Step-. This could be Adjustment, Open, Raise, Lower, or Close.
Activity for Verification Flow	An activity at the class of the step that calls a verification flow. If blank, the system

Field	Description
	automatically bypasses verification. One activity delivered with the product specifies a verification flow based on amount values. The system also has four verification flows (no verification required, one level, two levels, or three levels of verification required), which can be used, based on the amount of the step.
Modelto Create Step	A data transform rule name at the class of the step. If specified, the system creates the step page based on the model. The step rules in the PegaAcct ruleset do not use a model.
Validate Activity	Reserved for future use.
Likelihood	<p>An integer between 0 and 100 used to preselect the most likely step rule (the one with the highest number) during the manual flow actions that create accounting steps.</p> <p>The flow actions find all step rules that apply to the flow action. For example, a flow action of Open Payable on an encoding error finds all instances of Rule-PegaAcct-Financial-Payable for a specific class (and its parent classes) with an action of Open, and presents them in a selection box so you can choose which step to use. The default selection is taken from the rule in the list with the highest likelihood value.</p>

Field	Description
Transactions tab	
Group	Specifies if this transaction is part of a group of transactions so that only one transaction in the group is used when the accounting is created. In manual processing, you select which transaction from the group to choose (the Label property is displayed in a selection box so that you can choose only one transaction). In automatic processing, the first transaction in the group is selected, so the order is significant.
Label	This value appears in the window during the processing of manual accounting creation flow actions, so you can identify the transaction.
DR / CR	Indicates whether the transaction is a debit or a credit. The order of the transactions in the rule is the order in which they are displayed during manual accounting creation flow actions. Establish an order and follow it throughout all step definitions.
Role	A unique string identifier for each transaction in this step. Its value is placed into a Role property in the transaction of the step, and may be used to find or report on the transaction.
Txn Class	The class of the page that holds the transaction data; specify a class name that is

Field	Description
	a concrete class that is a child of class PegaAcct-Txn-.
Pymt Type	Name of Rule-PegaAcct-Financial-PaymentType rule to use for this transaction. The payment type rule associate correspondence with the transaction.
Default Account From	Party: The value comes from the Party role in the work object (The role is specified in the DefaultAccountValue property.) Chart of Accounts: The value comes from a specified internal account. The value of the DefaultAccountValue property is used in a lookup to the FinChartOfAccts map value rule at the class of the work object; this returns a key to an Internal Account record. A second map value rule is used in the class of the work object to determine the actual party class in which this key should be looked up. It finds the party class by looking up the InternalAccount value in the FinWorkPoolClasses map value rule.
Account From Value	Specifies a party role if the Default Account From value is a Party. Specifies an internal account name (as defined in the FinChartOfAccts map value rule) if the Default Account From value is Chart of Accounts.

Field	Description
Protect Account?	If selected, the Account Name and Number fields are protected, and you cannot enter values during manual accounting creation. If not selected, you can modify the values in these fields during manual accounting creation.
Default Amount From	<p>This specifies where to acquire the amount value for this transaction:</p> <p>Property:use the value of the property specified in the DefaultAmountValue property.</p> <p>Literal: use the value of the DefaultAmountValue property.</p>
Amount From Value	<p>Specifies a property reference if DefaultAmountFrom is a property. A reference can be any fully-qualified property reference from any page, but if the property reference is not dot-qualified, then it assumes the property will be found on the top level of the work object page.</p> <p>Specifies an actual amount number value if Default Amount From is Literal.</p> <p>Do not use quotes, \$, or commas (that is, specify 1000.00 and not \$1,000.00 or \$1000.00 or "1000.00").</p>

Field	Description
Default Currency	Specifies the currency code to default for the transaction amount.
Protect Amount?	If selected, the Amount field is protected, and you cannot enter values during manual accounting creation. If not selected, you can modify the value in the Amount field during manual accounting creation.
Default Value Date From	Specifies the value date for the From account.
Value Date From	Specifies the value date for the From value.
Value	
Protect Value Date?	If selected, during manual accounting creation, you cannot input a value in the Value Date field. If not selected, you can modify this value during manual accounting creation.
Amount From Value	<p>Specifies a property reference if DefaultAmountFrom is a property. A reference can be any fully-qualified property reference from any page, but if the property reference is not dot-qualified, then it assumes the property will be found on the top level of the work object page.</p> <p>Specifies an actual amount number value if Default Amount From is Literal. Do not use quotes, \$, or commas (that is, specify</p>

Field	Description
	1000.00 and not \$1,000.00 or \$1000.00 or "1000.00").
Default Currency	Specifies the currency code to default for the transaction amount.
Protect Amount?	If selected, the Amount field is protected, and you cannot enter values during manual accounting creation. If not selected, you can modify the value in the Amount field during manual accounting creation.
Default Value Date From	Specifies the value date for the From account.
Value Date From	Specifies the value date for the From value.
Value	
Security tab	
Name	Optional. Enter the names of any named privileges where access to the privilege allows this step to be executed.
Class	Optional. Enter the class of privileges that you specify in this window.
When Name	Optional. Enter a When rule that must be evaluated as true to allow this step to be executed. Most step rules in the application use When rules to allow access to the step only if the Party Role referenced in the step exists in the work object.
History tab	

Field	Description
Usage / Description	Full description of the instance and information on its usage.

Naming conventions for accounting step rules

When working with accounting step rules, it is a good practice to adopt a naming convention to define the Type property of the step. The name forms part of the key to the step rule and typically describes the type of account being processed or which parties are being debited and credited, or both. In addition, the name appears in accounting tabs, flow action drop-down lists, and reports.

The objective is to create a name that immediately communicates what is going on when the step is selected or processed, or both, to facilitate selection and understanding when forms and instructions are presented.

The following are some examples of Step rule naming conventions:

Step name	Rule	Description
ProvideProvisionalCRAcctOwner	Rule-PegaAcct-Financial-Receivable with action of Open	Credits the Account Owner and debits receivable marking the transaction as a provisional credit.
WriteOffReceivable	Rule-PegaAcct-Financial-Receivable with action of Close	Closes (debits) the receivable and writes offs the remaining balance.
StandardFee	Rule-PegaAcct-Financial-Adjustment with action of Adjust	Debits the customer for a standard fee, credits a fee account.

Using automatic processing flows and flow actions

The accounting straight-through processing (STP) flows and flow actions are designed and structured to process accounting without manual intervention. STP accounting supports the following:

- Adjustments with actions of Adjust and WriteOff.
- Suspense with actions of Open, Raise, Lower, Close, Offset, and WriteOff.

Automatic flows require a parameter used to specify which rule to use. The parameter is a name property. Referencing that parameter, the flow determines the other key parts (rule class, action, and work object class).

Manual flow actions can also determine the rule class, action, and work object class, and then display the **Short Description** from every step rule that can be used by the flow action (including security validation). By default, the product selects the transaction information from the rule with the highest likelihood value, and lets you select any other valid step rules from a selection box that includes the rules and their short descriptions.

When you create new step rules, you must update flows to call the rules, or add rules to give you more choices than those defined for automatic flows. In this case, the Likelihood value becomes significant.

- **Write-off**
- **Aged write-off**

Write-off

Accounting supports two types of write-offs:

- An automatic write-off of small dollar differences.
- Write-off of aged suspense entries
- **Small dollar write-off**

- How write off thresholds work

Small dollar write-off

There are two types of automated write-offs for small dollar differences:

- **On Entry** – Within a flow, a write-off threshold that if satisfied, automatically offsets a suspense debit or credit entry with an entry to the write-off account or adjusts a party's account with a corresponding entry to a write-off account.
- **On Lower** (draw-down) of an open suspense balance – A write-off threshold that if satisfied, automatically closes the suspense entry by writing off the remaining difference following a drawdown of the outstanding balance.

In each case, a small dollar threshold is defined for the highest dollar amount that you can write off. When you install Accounting, the On Entry threshold is set to \$50 and the **On Lower** threshold is set to \$20. The processes are enabled for all financial workflows and customer initiated non-financial workflows on entry and open suspense payable and receivable entries for drawdown.

For initial deployment, update the thresholds and workflows to reflect your institution's requirements.

Accounting records affected are:

Map value rules

- Map value rules
- WriteOffOnLower

The following screen shot shows the threshold rule for the write-off on entry.

Bank number	Default
Default	50.00

The following screen shot shows the rule for the write-off on lower.

A screenshot of a Pega application window titled 'FinWriteOffOnLower'. The window has a header with tabs like 'Matrix', 'General configuration', 'Parameters', etc. Below the header is a toolbar with buttons for 'Configure columns', 'Configure rows', 'Delete Columns', 'Import', and 'Export'. The main content area contains a table with one row. The first column is 'Bank number' and the second column is 'Default'. The 'Default' value is set to '20.00'.

Note: To remove the write-off evaluation from a workflow, delete the SmallDollarWriteoff flow element from the flow diagram of the workflow. To add to a diagram, the recommended practice is to perform the evaluation before requiring the system to perform additional work that will not be necessary if the write-off is performed. To deactivate the WriteOffOnLower feature from suspense processing, set the default value to 0.00.

How write off thresholds work

Two write-off thresholds are used:

- On Entry
- On Lower

On Entry

The **FinWriteOffOnEntry** flow determines the write-off threshold by using the map value rule, which is specified as a parameter to the flow. The default MapValue rule is **FinWriteOffOnEntry** and is set to \$50.

The flow uses the **FinWriteOffEvaluate** activity which calls the **FinGetWriteOffThreshold** activity to get the threshold amount. The activities then use the map value to return the threshold amount which is based on the input of a BankNumber located in the work object property .InternalBank. All the rules are at the work object class.

If the Amount property, also passed in as a parameter, is less than the threshold amount then it processes an Adjustment step rule named in the StepRule parameter with an action of **Writeoff**. To enable this function, you must create an Adjustment step rule in your work object class (or parent) with an action of **WriteOff** and a type that matches the value passed to the **WriteOff** on Entry Flow.

Add this flow to your workflows for financial cases, which require this functionality.

On Lower

As part of posting in the FinWriteOffAfterPosting flow, either after a lower or partial offset, the FinWriteOffOnLower flow is executed and it uses the FinWriteOffOnLower map value which is set to

\$20, and checks if the remaining suspense balance is below the threshold in the map value rule. If the balance is below the threshold, then it processes either a specific Payable or Receivable step rule of action WriteOff. The step rules are named either WriteOffPayable or WriteOffReceivable which are pre-defined step rules. These rules use the symbolic name for Chart of Account lookups of WriteOff, and either Payable or Receivable. Either of these values must exist in the Chart of Account map value called FinChartOfAccts or you should modify these step rules to reference the correct symbolic name for your accounts.

This functionality is a part of the standard posting process and is a functionality that is automatically on for suspense accounting. You should ensure that the configuration for referencing the Chart of Accounts is correct.

Aged write-off

Accounting supports the aging, automatic scheduling, and processing of open receivable (debit) and payable (credit) write-off suspense entries associated with work objects.

Service level rules define the schedule for aging open payable (credit) and receivable (debit) suspense entries in the system. Different aging schedules can be defined for

receivables and payables in either business or calendar days. The accounting application ships with a sample configuration for these aging parameters.

When a flow opens a suspense entry, it references a rule to calculate the scheduled resolution date of the receivable or payable and begins the aging process. The scheduled resolution date for a case is calculated using the outstanding value at the time of creation and the aging service level rules.

Service level rules also define the schedule for the actual writing off the aged entries. When a suspense entry is opened, the entry is immediately assigned to the FinAging work queue where it ages until the entry is either closed or until the service level expires. If the service level expires, the application transfers the entry to the **FinWriteOff** work queue. To actually write off entries from the work queue, you call the **FinBatchWriteoff** activity.

Operators, with privileges, can reschedule suspense entries for an earlier write-off by transferring them directly to the **FinWriteOff** work queue during the research process. Conversely, you can postpone the write-off by selecting the entry from the **FinWriteoff** work queue and using the reschedule option to select a new service level for the entry and automatically transfer it back to the aging work queue.

- **Referenced by**
- **Format**
- **Accounting records affected**

Referenced by

- FinWriteOffAfterPosting flowrule
- Selection box list of SLAs on option to reschedule write-off from the Accounting Aging and Accounting WriteOff work queues



Note: The Aged Write-off process is triggered in the FinStepPostProcessing flow.

Format

- Service Level Agreements: numeric in hours, days, minutes, and seconds
- Date calculation: numeric

Accounting records affected

- Service level rule – Defines an aging schedule in either calendar or business days. Referenced in the write-off map value rules and the reschedule option on the write-off work queues.
 - FinWriteOffAfter30
 - FinWriteOffAfter60
 - FinWriteOffAfter90
- Map value rule – References the service level that is applied to set the scheduled write-off date of the suspenseentry.
 - FinPayableWriteOffSLA
 - FinReceivableWriteOffSLA
- HTML rule – Creates the list of service level rule options for rescheduling the write-off of a suspense entry.
 - FinWriteOffReschedule

The following screen shot shows the form you use to set up a service level. This is the service level rule for Financial Write-off after 30 calendar days.

Service Level Agreement: Wait 30 days before Aged Write Off [Available]

CL: Work- ID: FinWriteOffAfter30 RS: PegaAccounting:08-01-01

General Specifications History

Start of service level

Initial Urgency

Assignment Ready Immediately

Service level definitions

Calculate service levels Interval from when assignment is ready

Goal

Days Hrs Mins Secs
30 0 0 0

Amount to increase urgency 0

Actions	When
Run Activity	Edit <input type="button" value="Delete"/>
+	

Time interval starts when the associated assignment (or work item) is created

Only calculate using business days

Deadline

Days Hrs Mins Secs
30 1 0 0

Amount to increase urgency 0

Actions	When
Run Activity	Edit <input type="button" value="Delete"/>
+	

Time interval starts when the associated assignment (or work item) is created

Only calculate using business days

Passed deadline

Limit passed deadline events to 1

Days Hrs Mins Secs
0 0 0 0

Amount to increase urgency 0

Actions	When
Select Action...	Edit <input type="button" value="Delete"/>
+	

Time interval starts when the deadline is reached

Only calculate using business days

The following screen shot shows the map value rule that you complete to set up the payable write-off criteria, referencing a service level.

Map Value : Returns the SLA for Aged Write Off [Available]

Work- #FinPayableWriteOffSLA PegaAccounting:08-01-01

Matrix General configuration Parameters Pages & Classes Test cases Specifications History

Configure columns Configure rows Define conditions Define components Export Import

Outstanding Balance Default

Default FinWriteOffSLA

The following screen shot shows the map value rule that you complete to set up the receivable write-off criteria, referencing a service level. If <\$100 – write-off in 30 days, <10000.00 write-off in 60 days, otherwise write off in 90 days.

This screenshot shows the 'Map Value' configuration interface for a 'FinWriteOffCriteria' service level. The map defines rules based on outstanding balance:

- <= 100.00: FinWriteOffAfter30Days
- <= 10000.00: FinWriteOffAfter60Days
- >= 250000.00: FinWriteOffAfter90Days
- Default: FinWriteOffAfter90Days

The following screen shot shows the HTML and the When statement that indicates the service level rules to be displayed as reschedule options.

This screenshot shows the 'Section: Fin WriteOff Reschedule' configuration interface. It displays a single reschedule option named 'Option 1' under the 'Service Level' section.

Note: Define your service levels first. There should be a service level record for every possible schedule. Then, define the decision maps for payables and receivables that reference the service levels established on the opening of the suspense entry.

In some situations, you may need to create more than one decision map. For example, if a receivable write-off is scheduled for 90 days on the open of a suspense entry, but managers may have the option of postponing for an additional 30 days, you must create two maps: one for 90 days and one for 30 days.

Duplicate and offset processing

Accounting lets you automatically flag items as potential duplicates, financial offsets, or item offsets based on search and scoring rules that are able to factor item amounts, work dates, case types, as well as party, accounting, and case data elements into the evaluation.

It also supports automatic background resolution of duplicates and offsets using confidence-level testing. When activated, this feature automatically resolves a duplicate or offset that meets a confidence threshold without manual operator review and evaluation.

- [Duplicate scoring rules](#)
- [Item offset search](#)
- [Financial offset search](#)
- [Routing assignments to work queues](#)

Duplicate scoring rules

The application supports the search for, scoring, linking, and resolution of potential duplicate cases. The results are based on criteria that you define in duplicate search and scoring rules. You can define different search criteria at the case level for different case types.

Duplicate searching combines parameters and instructions from three sources that may be defined at the case type or general class level:

- **AppSetDuplicateClasses** – An activity that defines the data against which the search to extract duplicate candidates is performed.
- **AppSearchDuplicates** – A flow with an embedded activity that defines the initial search (in Java Steps) and extraction threshold criteria.
- **DuplicateScore** – An instance of Rule-PegaAcct-Search-Score that defines the scoring parameters and the confidence threshold.

Item offset search

Item offset search combines parameters and instructions from three sources which may be defined at the case type or general class level:

- **FinSetOffsetClasses** – An activity that defines the data against which the search to extract candidates is performed.



Note: This record is referenced by the financial and the non-financial offset search.

- **FinNonFinancialOffsetSearch** – A flow with an embedded activity that defines the initial search (in Java Steps) and extraction threshold criteria.
- **Rule-PegaAcct-Search-Score** – An instance of this rule that you create to define the scoring parameters and the confidence threshold. You can use the DuplicateScore instance as the model from which to copy.

Financial offset search

Financial offset searching combines parameters and instructions from three sources that you can define at the case type or general class level:

- **FinSetOffsetClasses** – An activity that defines the data against which the search to extract offset candidates is performed.
- **FinFinancialOffsetSearch** – A flow with an embedded activity that defines the initial search (in Java Steps) and extraction threshold criteria.
- **Rule-PegaAcct-Search-Score** – You create an instance of this rule to define the scoring parameters and the confidence threshold. You can use the DuplicateScore instance as the model to copy from.

Routing assignments to work queues

The accounting flows route assignments to the following work queues:

- **FinAging** – Holds and age suspense entries for write-off.
- **FinWriteOff** – Holds suspense entries that passed aging criteria and are ready to be written off.

- **FinVerification** – Holds accounting entries that require verification prior to posting.
- **FinCutoff** – Holds accounting entries that cannot be processed due to a cutoff time (only used for standard cutoff type of Hold).

The routers in the accounting flows use symbolic names and lookup the work queue to route to via the FinWorkbaskets map value rule.

Verifying accounting functionality

After installing Pega Foundation for Financial Services, perform the following steps to verify that the accounting functionality is working.

1. Log in with the following credentials:

Username: Jeff.Reynolds@uplusfs.com

Password:install

2. From the New menu, select Create new accounting. The initial screen, Accounting Test Flow, appears.
3. Complete the following required fields and click Create.
 - Amount and Currency Code
 - Value Date (defaults to today)
 - Account Number
 - First Name
 - Last Name
 - Email Address
4. Click the Other actions menu to view the available actions.

Other accounting actions are available under the Add Work option.

5. Select Open a Payable (CR) to view the Open Payable options and defaults.
6. Click Submit.

7. Click the Suspense Count tab and view the suspense details. This concludes the basic verification steps.

Pega Foundation for Financial Services requirements guide

- [About this document](#)
- [Overview](#)
- [Using Requirement rules](#)
- [Using Requirement set rules](#)
- [Passing data between a requirement and its parent case](#)
- [Attachments in requirements](#)
- [Requirement rule form help](#)
- [Requirement set rule form help](#)

About this document

This document describes the core functionality available in the Pega® Foundation for Financial Services. For information about the Requirements Portal, see the Requirements Portal – Configuration Guide on the Pega Foundation for Financial Services product page.

- [Intended audience](#)

Intended audience

This guide is intended for developers who are customizing these rules during initial product design and deployment.

Overview

The Pega Foundation for Financial Services helps organizations define and manage multi - stage, complex document collection processes.

Documents are defined that will be required based on the purpose. For example, an appraisal report, passport, or signed application.

Applicability logic determines when documents are required. For example, determining which customer identification programs need to be met when onboarding a new company.

Satisfaction logic determines that the required documents have successfully obtained. For example, defining the combination of documents that meets a jurisdiction's customer identification program.

While running a case such as a mortgage application or onboarding a customer, interfaces are provided for you to manage sourcing, reviewing, and approving documents.

Two custom rule types have been created that support the up-front designing of applicability and satisfaction logic as well as managing individual cases as they are run:

Requirements: rules support identifying that the requirement is applicable along with the documents that need to be satisfied. The Requirement object (Rule-Requirement) is created at runtime and linked to a stage in the processing flow of the parent case.

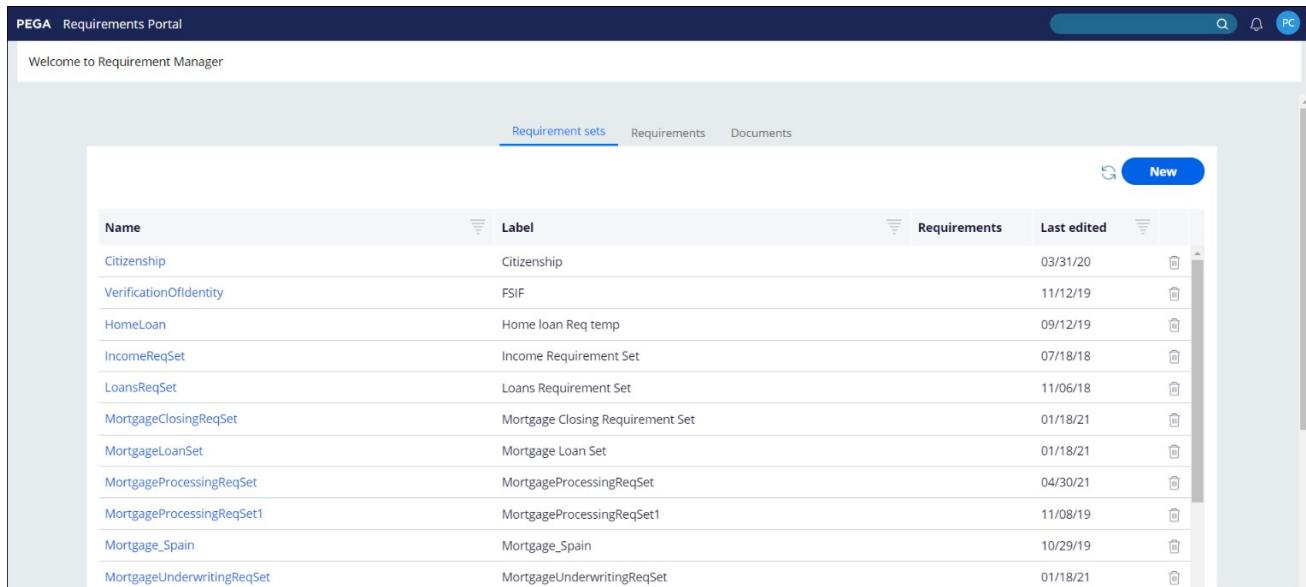
Requirement Sets: rules that support identifying that the Requirement Set is applicable along with the Requirements that need to be satisfied. The Requirement Set rules (Rule- RequirementSet) are used to group Requirement rules together and apply them to a parent case. At any time in the parent case process, one or more Requirement Sets can be processed for the parent case, creating linked Requirements.

Using Requirement rules

- Selecting Requirement objects
- Document-based requirements
- Configuring document entries in requirement rules
- Unguided validation
- Guided validation
- Overall requirement validation
- Flow-based requirements

Selecting Requirement objects

The Requirements Portal enables users to quickly find, edit, or create requirement items and is organized using the following tabs: Requirement Sets, Requirements, and Documents:



The screenshot shows the PEGA Requirements Portal interface. At the top, there's a dark header bar with the PEGA logo and the text "Requirements Portal". Below it, a white header bar says "Welcome to Requirement Manager". Underneath, there are three tabs: "Requirement sets" (which is selected), "Requirements", and "Documents". A blue "New" button is located at the top right of the main content area. The main content area is a table with columns: "Name", "Label", "Requirements", and "Last edited". The table lists various requirement sets, each with a "Edit" icon. The data in the table is as follows:

Name	Label	Requirements	Last edited
Citizenship	Citizenship		03/31/20
VerificationOfIdentity	FSIF		11/12/19
HomeLoan	Home loan Req temp		09/12/19
IncomeReqSet	Income Requirement Set		07/18/18
LoansReqSet	Loans Requirement Set		11/06/18
MortgageClosingReqSet	Mortgage Closing Requirement Set		01/18/21
MortgageLoanSet	Mortgage Loan Set		01/18/21
MortgageProcessingReqSet	MortgageProcessingReqSet		04/30/21
MortgageProcessingReqSet1	MortgageProcessingReqSet1		11/08/19
Mortgage_Spain	Mortgage_Spain		10/29/19
MortgageUnderwritingReqSet	MortgageUnderwritingReqSet		01/18/21

Document-based requirements

You can define requirements assuming that a requirement is satisfied by the receipt and validation of a certain document or set of documents. The documents needed to satisfy a requirement may be any combination of the following:

- System-produced documents, for example, a loan application prefilled with the borrower's information.
- Documents sourced outside the system, for example, an applicant's tax return from the previous year or a form produced by an outside form-generation system.

After a document has been completed by the applicable parties, you can scan it into a document repository and then link the image to one or more requirements that call for it. After you associate an image with a requirement document entry, you can validate it as described later in this chapter.

- **Document types**

Document types

You can define three types of documents in the library: manual document library entry, binary document library entry, and eForm document library entry.

Manual document library entry

A manual document entry defines a document that is sourced from outside the parent case. The system does not generate any forms to fulfill this kind of document requirement.

It indicates that an image of a particular document must be delivered to meet the requirement. An example of this type of document requirement is the applicant's most recent tax return, or a passport copy. A manual document in the library does not require keys since it does not cause any form generation.

Binary document library entry

A binary document entry defines a PDF document that is generated by the system based on a RuleFile-Binary Pega Platform rule, which is a binary image. This allows for generation of a static document. It does not allow data from the parent case to be included in the form. An example of this document type is a blank application form or standard disclosure documents. A binary document in the library has key fields that identify the Rule-File-Binary used to generate the document.

EForm document library entry

An EForm document entry defines a PDF document that is generated by the system based on a Rule-File-eForm/Rule-Map-eForm Pega Platform rule pair. Together, these rules define a PDF form and the rules for mapping data from the parent case into the document. This allows for generation of a document that is prefilled with data from the parent case. An example of this document type is an application form that has been prefilled with applicant data. An EForm document in the library has key fields that identify the Rule-Map-eForm used to generate the document.

Configuring document entries in requirement rules

A Requirement rule can include any number of document entries that are processed to determine which optional and required document entries apply to a requirement at runtime. Each document entry in the rule (as well as data in the parent case at runtime) determines whether and how that entry is processed at runtime.

Configuration

The screenshot shows the Requirements Rule configuration interface for document W2. At the top, it displays the requirement ID W2 and its subject tags. The main area is divided into several sections:

- Requirement:** Includes fields for Document Only/No Requirement, Type of Requirement (set to Guided Document Validation), Verification Section, Matching Activity Name, Days Until Stale (25), Validation Class, Processing Flow Name, and Processing Instructions.
- Applicability Logic:** Fields for Applicability Logic, Satisfaction Logic, and No. of Documents.
- Documents for Verification:** A detailed section with fields for Label, Document ID (W2), Description (W2), Applies when, Reuse Existing, Generate EForm before case Creation, Send Email, EForm Satisfies requirement?, PartyAppliesWhen, Valid Days, Verification UI Section (VerifyW2), and Verification Validation (ValidateW2).

The document entry settings, described below, control whether a document entry is applicable to a particular parent case at runtime. (See *Appendix A* for a detailed description of all the fields on the Requirement Rule form.)

Document ID – Identifies the specific type of document.

Completion Required – Indicates whether the document must be completed and In Good Order before this requirement can be satisfied.

Applies-When – Identifies a When rule that can determine whether this document entry is applicable to a particular parent case. For example, a document entry may be applicable in some states, but not in others. The When rule uses any data related to the parent case that is available at runtime, and can be as simple or complex as necessary.

Reuse Existing – Indicates whether the application should search at runtime for a document image already in the system that satisfies this entry and link it to this requirement if one exists. Otherwise, it does not search for an existing image.

Party-Applies-When – Identifies a When rule that determines whether this document type is applicable to a particular party on the parent case.

Unguided validation

Any mandatory requirement document entry must be validated before the requirement can be resolved as satisfied. The simplest type of validation is *unguided*. This approach provides no specific guidance about how to validate the document. It assumes that you are qualified to determine if a document is valid.

Configuration



In the Requirement rule above, the document is required but the **Verification Section** and **Validation** entries are blank. This indicates that the document validation will be unguided.

Runtime results

At runtime, you view both the image of the document and the validation screen for validating the document.



Indicate whether you find the document to be *In Good Order*. Once you do, the status for the document entry is changed to *In Good Order*, and the document entry is satisfied.

The screenshot shows a 'Verify Requirement' screen for a 'New personal loan (AA-87002)'. On the left, under 'Verify Requirement', there's a section for 'HUD-1 Mortgage Closing Statement' with a file icon, a checkbox for 'Upload another document', and links for 'Version history' and 'Comments'. Below this is a note: 'Review the document for the following criteria and provide additional comments as needed'. A checked checkbox says 'Does the HUD statement look authentic?'. There are fields for 'Notes:' (a large text area) and 'Comments' (a smaller text area with placeholder 'Type your comment and press Enter to confirm.'). At the bottom are 'Cancel' and 'Submit' buttons. On the right, a vertical sidebar titled 'Validate' shows a step '2 Verify requirement' with a link 'Validate Overall Requirement'.

Guided validation

Guided validation is a more robust approach to validation. It does not assume that you have the knowledge necessary to determine if the document is valid. A custom validation screen appears for data entry at document validation time. The data you enter could be as simple as selecting check boxes in a list, or as complicated as entering specific data (amounts, dates, and so on) from the document into the validation screen. Once you enter all the data into the validation screen, the system runs a validation rule to determine if, based on the data you entered, the document is satisfactory for this requirement.

Configuration

Requirements: W2 [Available]

PegaReq Data-ReqVerify - ID: W2 RS: PegaFSSample:08-02-01

Type of Requirement: Guided Document Validation

Requirement Section:	Validation:	Matching Activity Class:
Matching Activity Name:	Processing Flow Class:	Processing Flow Name:
Days Until State:	Processing Instructions:	
Applicability Logic:	Satisfaction Logic:	No. of Documents:

Documents for Verification

Label:	Document ID: W2	Description: W2	Applies when:
	Attachment Category:	<input type="checkbox"/> Separate Doc for Each Party	<input type="checkbox"/> Reuse Existing
	<input type="checkbox"/> Generate EForm before case Creation	<input type="checkbox"/> Send Email	<input type="checkbox"/> EForm Satisfies requirement?
	PartyAppliesWhen:	Verification UI Section: VerifyW2	Verification Validation: ValidateW2
	Valid Days:	Image Search Params:	

In the Requirement shown above, the W2 is required, and its validation is guided by two entries:

- The **Verification Section** VerifyW2 appears to you at validation time to enter information about the W2 being validated.

You must define the section in the same class as the Requirement rule or in an ancestor class.

Section: VerifyW2 [Available]
 CL: PegaReq-Data-ReqVerify- ✓ ID: VerifyW2 RS: PegaFSRequirements:08-02-01
 Design Settings Parameters Pages & Classes HTML Specifications History

Structural ✓ Data capture ✓ Pickers ✓ Action ✓ Data display ✓

- The **Validation** ValidateW2 is run against the data that you enter to determine if the document is valid for this requirement.



Note: You must define the validation in the same class as the Requirement rule or in an ancestor class.

Validate: ValidateW2 [Available]
 CL: PegaReq-Data-ReqVerify- ✓ ID: ValidateW2 RS: PegaFSRequirements:08-01-01
 Validate Input Pages & Classes Specifications History

Expand all Collapse all Default Validation

PROPERTY	*Req Conditions	Edit
.AppliesToPrior2Yrs	IF value is "false" THEN display message: TwoYearTerm	
.NoUnderstatedIncome	★ IF value is "false" THEN display message: IncomeByBorrower	Edit
ADDITIONAL VALIDATION		

Runtimes results

At runtime, you view both the image of the document and the validation screen for validating the document. In the case of this guided validation example (as configured above), the screen looks like the one below.

Validate Required Documents

Description	W2
Attached Documents	View
Parties	Ryan Roberts
Please carry out document validation instructions below	
<input type="checkbox"/> Applies to prior 2 years	
<input type="checkbox"/> All reported income is disclosed by borrower or is verified to be reduced	
Gross Wages from Box 1	
Tips from Box 7	
Overtime, Commissions, Bonus, and Other not identified separately	

Status: Pending In Good Order Review

Valid End Date: _____

2010 W-2 and EARNINGS SUMMARY					
This blue Earnings Summary section is included with your W-2 to help describe portions in more detail. The reverse side includes general information that you may also find helpful.					
1. The following information reflects your final 2010 pay stub plus any adjustments submitted by your employer.					
Gross Pay	72017.70	Social Security Tax Withheld	4189.92	N.J. State Income Tax	2477.36
		Box 4 of W-2		Box 17 of W-2	
Fed. Income Tax Withheld	12201.41	Medicare Tax Withheld	979.91	STATE/TAX	274.72
Box 4 of W-2		Box 6 of W-2		Box 14 of W-2	
2. Your Gross Pay was adjusted as follows to produce your W-2 Statement.					
Wages, Tips, other Compensation	72,017.70	N.J. State Wages, Tips, Etc.		Social Security Wages	
Box 1 of W-2		Box 16 of W-2		Box 5 of W-2	
Gross Pay	72,017.70	72,017.70	N/A	72,017.70	72,017.70
Less Net Subject	4,438.21	4,438.21	N/A	4,438.21	4,438.21
Reported W-2 Wages	67,579.49	67,579.49	N/A	67,579.49	67,579.49
3. Employee W-4 Profile. To change your Employee W-4 Profile Information, file a new W-4 with your payroll dept.					
RYAN ROBERTS 1 MAIN ST ARLINGTON, MA, 02474					
Social Security Number: [REDACTED] Tax Filing Status: [REDACTED] Exemptions/Allowances: [REDACTED]					
FEDERAL: 0 STATE: 0 Table A					

The screen includes requests for five specific pieces of information from the W-2, as shown above.

In the example, the operator has entered four of them, but neglected to note whether the W-2 is within two years. The system responds with a message that the document is not valid because the operator has not certified that the document refers to one of the prior two years.

New personal loan (AA-87001)

W2

Upload another document
 fst_Sample_Bank
[Version history](#) [Comments](#)

Review the document for the following criteria and provide additional comments as needed

Applies to prior 2 years W2 has to apply to prior two years
 All reported income is disclosed by borrower or is verified to be reduced Understated Income by Borrower

Gross wages from box 1

Tips from box 7

Overtime, Commissions, Bonus, and Other not identified separately

Comments
 Type your comment and press Enter to confirm.

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

After the operator receives a good W-2 and certifies that it is in fact from the current year, the system revalidates the document.

Document Requirements

[Validate Overall Requirement](#)

Pay Stub Pending Receipt [Awaiting Receipt\(1\)](#)

Ryan Roberts

Manage Documents

W2 In Good Order Document Reviewed(2)

Ryan Roberts

[Manage Documents](#)

The system determines that the document is now valid and sets its status to *In Good Order*, as shown above. This completes the guided validation of this document.

Overall requirement validation

In addition to performing guided validation at the individual document level, the application provides a parallel mechanism at the level of the overall requirement. This allows the system to collect data and perform validation across multiple documents or collect data and perform validations that are not related to any document. For example,

you can use this capability to ensure that certain data items collected from one document match data items from a different document.

Configuration

The screenshot shows the Pega Requirements configuration interface for a 'Mortgage Personal Identification' requirement. The 'Type of Requirement' is set to 'Guided Document Validation'. The 'Verification Section' is labeled 'VerifyMortgageID'. The 'Validation' step uses 'ValidateID' with a 'Matching Class' of 'MortgagePersonalIdentification'. The 'Processing Flow' step uses 'ProcessFlowName' with a 'Name' of 'VerifyMortgageID'. The 'Processing Instructions' step contains the note: 'Ensure ID matches applicant and across several documents.' Below this, there are sections for 'Applicability Logic', 'Satisfaction Logic', and 'No. of Documents'. The 'Documents for Verification' section lists two entries: one for 'Passport' and one for 'StateID', each with its own validation logic and processing steps.

In the Requirement shown above, there is an overall Requirement validation in addition to the document validations. This overall Requirement validation is guided by three entries:

The Overall Verification Section *VerifyMortgageID* is presented at validation time so that you can enter information about the information being validated.

You must define the section in the same class as the Requirement rule or in an ancestor class.

Section: VerifyMortgageID [Available]
CL: PegaReq-Data-ReqVerify- **ID:** VerifyMortgageID **RS:** PegaFSRequirements:08-02-01

Save as Actions Private edit X

Design Settings Parameters Pages & Classes HTML Specifications History

Dynamic Layout (Stacked) - 1
Verify Requirement
Repeating Dynamic Layout (Default) [pyWorkPage.ReqValidateDocs of Class PegaReq-Data-ReqVerify] - 1.1

Passport name _____ State ID name _____
Passport DOB _____ State ID DOB _____

Dynamic Layout (Simple list) - 1.2

Dynamic Layout (Stacked) - 1.3

Data provided for review purposes only. Data must be corrected at document-level verification.

Validate: ValidateID [Available]
CL: PegaReq-Data-ReqVerify- **ID:** ValidateID **RS:** PegaFSRequirements:08-01-01

Save as Actions Private edit X

Validate Input Pages & Classes Specifications History

Expand all Collapse all Default Validation

PROPERTY

- .MortgagePassportName
- .MortgagePassportDOB

***Req Conditions**

- IF .MortgagePassportName != .MortgageStateIDName
THEN display message:
NamesMustMatch
- IF .MortgagePassportDOB != .MortgageStateIDDOB
THEN display message:
BirthdatesMustMatch

ADDITIONAL VALIDATION

The Processing Instructions provide guidance to the operator regarding how to process the requirement.

Runtime results

In this example, the operator has already validated the Passport and State ID documents , and has incorrectly entered the Date of Birth on the State ID. When the operator performs the overall Requirement validation, the operator is presented with the validation screen for validating the document. In the case of this guided validation example (as configured above), the screen looks like the one below.

The screenshot shows a 'Validate Overall Requirement' dialog box. At the top, there's a header bar with the title 'Validate Overall Requirement' and a close button (X). Below the header, the section 'Verify Requirement' is displayed. It contains two sets of data side-by-side:

Passport Name	State ID Name
Ryan Roberts	Ryan Roberts

Passport DOB	State ID DOB
1/13/80	1/13/80

A note at the bottom of the dialog states: 'Data provided for review purposes only. Data must be corrected at document-level verification.' At the bottom right are 'Cancel' and 'Submit' buttons.

The operator is asked to *Ensure ID matches applicant and across several documents*, and is presented with the data as entered from the documents.

In the example below, the Date of Birth entries for the two documents do not match. If the operator tries to proceed, the validation rule indicates why the overall Requirement cannot be validated.

Errors:

- Passport Date of Birth: Birthdates must match

New personal loan (AA-1)

Validate overall requirement
Verify Requirement

Ryan Roberts

Passport name	State ID name
Ryan Roberts	Ryan Roberts
Passport DOB	State ID DOB
12/1/90	8/1/90

Data provided for review purposes only. Data must be corrected at document-level verification.

Ensure ID matches applicant and across several documents

The operator revalidates the documents to repair the birth date error. In this example, it is assumed that the mismatch was due to operator error rather than an actual mismatch on the documents. After the error is corrected, the operator can revalidate the overall requirement.

New personal loan (AA-2)

Validate overall requirement

Verify Requirement

Ryan Roberts

Passport name	State ID name
Ryan Roberts	Ryan Roberts
Passport DOB	State ID DOB
1/13/80	1/13/80

Data provided for review purposes only. Data must be corrected at document-level verification.

Ensure ID matches applicant and across several documents

The system determines that the overall Requirement is now valid and the requirement is completed by changing its status to *Resolved-Completed*.

Personal Loan Application (AA-410)

Personal Identification (RQ-1808)

Save **Other actions ▾** **Close**

Processing Instructions

Ensure ID matches applicant and across several documents

Document Requirements

* **Passport** In Good Order Document Reviewed (2) Manage Documents

No items

* **State ID** In Good Order Document Reviewed (2) Manage Documents

No items

Flow-based requirements

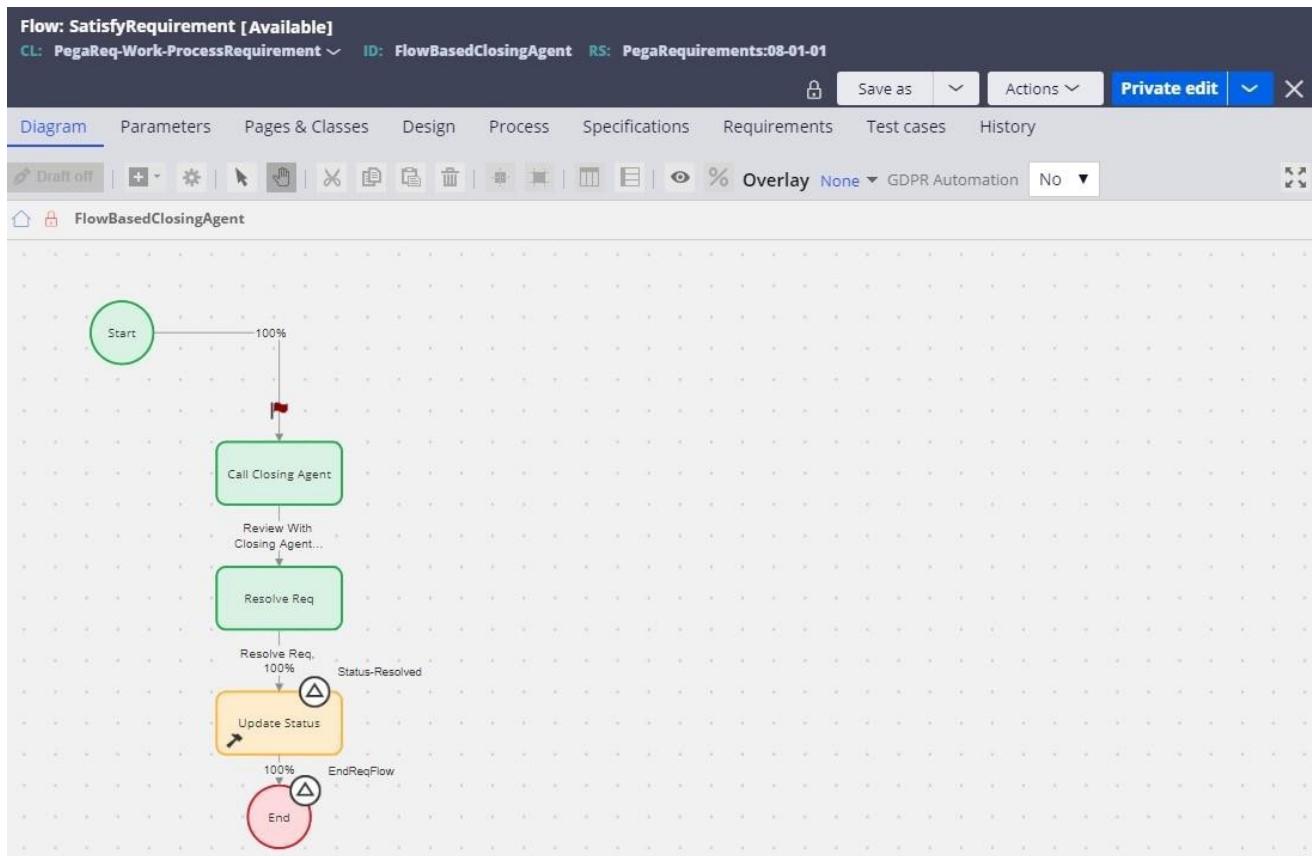
Some business needs involve more than document validation. They may involve performing business processes that include multiple steps and multiple users, external integrations, complicated process branching, and so on. In this case, the application launches a requirement on a process flow written specifically to meet the business need. This allows a requirement to support complicated business processes in addition to document validation.

Configuration

The screenshot shows the 'Requirements' configuration screen for a requirement named 'Closing Agent Contact'. The top bar displays the requirement name, its status as 'Available', and various navigation options like 'Save as', 'Actions', and 'Private edit'. The main area is divided into several sections:

- Type of Requirement:** Guided Document Validation
- Verification Section:** [Empty field]
- Validation:** [Empty field]
- Matching Activity Class:** [Empty field]
- Matching Activity Name:** [Empty field]
- Processing Flow Class:** PegaFS-Work-Requirements
- Processing Flow Name:** FlowBasedClosingAgent
- Days Until Stale:** [Empty field]
- Processing Instructions:** Confirm that the following activities have taken place.
- Applicability Logic:** [Empty field]
- Satisfaction Logic:** [Empty field]
- No. of Documents:** [Empty field]
- Documents for Verification:** A section labeled 'Label' with a '+' button for adding entries.

In the Requirement above, there are no document entries at all. Instead, the Processing Flow Class and Processing Flow Name fields indicate that when this Requirement rule is instantiated as a Requirement object, it will be launched on the following flow in order to be satisfied.



Runtime results

At runtime, the Requirement rule shown above creates a Requirement object and launches it on the specified flow. Then, you receive the first assignment of the flow.

New personal loan (AA-87002)

Closing Preparation Review

Closing agent contacted:

Closing Scheduled:

Borrower notified:

Flow Based Requirement

2 UnGuided Requirement Flow

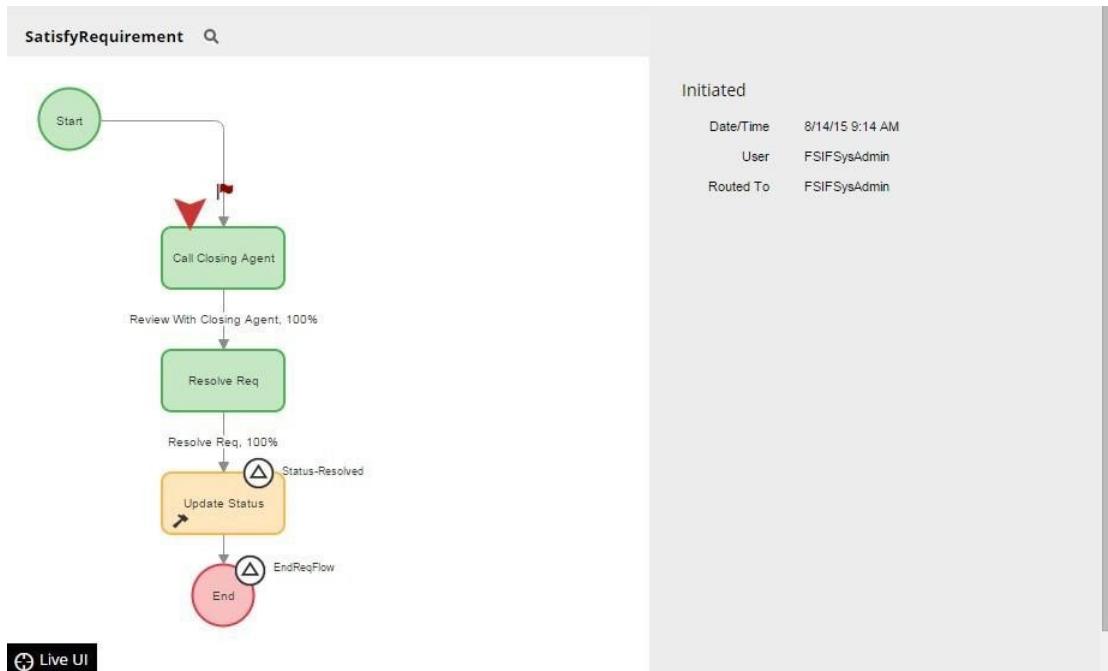
Pulse

Post • Start a conversation

No posts yet

Cancel Submit

The Where Am I screen shows that you are currently on the first assignment of the specified flow.



Using Requirement set rules

Requirement Set rules control which Requirement rules are applied to a particular parent case. In addition, all the requirements included in a Requirement Set are applied to a particular stage in the process flow of the parent case. A stage is a named point in the parent case where a checkpoint is inserted to ensure that all requirements applied to that stage have been satisfied before the parent case process continues.

- **Configuring requirement entries in requirement set rules**
- **Selecting a requirement set**
- **Configuring stage processing**

Configuring requirement entries in requirement set rules

A Requirement Set rule can include any number of requirement entries that are processed to determine which Requirement rules should be instantiated for a particular parent case and process stage at runtime. Each requirement entry in the Requirement Set rule (as well as data in the parent case at runtime) determines whether that requirement entry is instantiated at runtime. You should define the Requirement Set rule in the same class as the parent case or in an ancestor class of the parent case.

Configuration

RequirementSets: Mortgage Processing Requirement Set [Available]
 CL: PegaFS-Work-NewAccount ID: MortgageProcessingReqSet RS: PegaFSSample:08-02-01

Save as Actions Private edit X

Requirement Rules Parameters EForm Prep History

Applies when

Allow this requirement set to be manually added to work in process

Label	Class Name	Requirement Rule Name*	Applies When	Routing Activity	Overall SLA Activity	Stale Date SLA Activity
	PegaReq-Data-ReqVerify-	Personal Identification				
	PegaReq-Data-ReqVerify-	Mortgage Application				
+						
Satisfaction Logic						
ConditionalSuitabilityFilter						

The Requirement entry settings, described below, control whether a Requirement is applicable to a particular parent case at runtime. (See *Appendix B* for a detailed description of all the fields on the Requirement Set Rule form.)

Class Name – Identifies the class on which the Requirement rule for the entry is defined.

Requirement Rule Name – The name of the Requirement rule for this entry.

Applies When – Identifies a When rule that can determine whether this Requirement entry is applicable to a particular parent case. For example, a Requirement entry may be applicable in some states, but not in others. The When rule can use any data related to the parent case that is available at runtime, and can be as simple or complex as necessary. You should define this in the same class as the parent case or in an ancestor class of the parent case.

In the example above, the MortgageProcessingReqSet Requirement Set rule includes three Requirement entries. The first two are unconditionally instantiated and linked to the parent case at runtime. The third Requirement entry, for Proof of Residence, is

instantiated only if the When rule `PropertyIsInAZ` is *True*. This means that it is applicable only when the mortgage property in the application is in the state of Arizona (see sample When rule below).

Selecting a requirement set

At any point in the process flow for the parent case, Requirement Set rules can be processed to create and link new requirements to the parent case and apply them to new or existing stages in the parent case flow. This is done by calling the `GenerateMilestoneRequirements` utility activity. The input to this activity is a reference to a clipboard page of the `Data-RequirementSetList` class. It is a page list of entries of Requirement Set rules for the activity to process. Each entry in the list has four properties:

ReqSetClass – The name of the class on which the Requirement Set rule for this entry is defined.

ReqSetName – The name of the Requirement Set rule for this entry.

Milestone or Stage – The name of the milestone or stage to which requirements in this

Requirement Set should apply. If a milestone or stage with this name does not exist, it is created.

TargetDeadlineDate – This optional property sets a target completion date for the stage – a date by which all the requirements that apply to the stage should be satisfied. This date can be used for standard Pega Platform SLA processing.

In the Requirements functionality, a model based on the selected product to load the Data - RequirementSetList page is used for processing. During implementation, you can create these pages however you want – models, data tables, external integrations, PCF, and so on. The model for the FNMAConforming30YearFixedRateMortgage sample product in the application is shown below.

Action	Target	Relation	Source
• 1	.ReqSetListPage1.ReqSetList(1).ReqSetClass	equal to	"PegaFS-Work-NewAccount"
• 2	.ReqSetListPage1.ReqSetList(1).ReqSetName	equal to	"MortgageProcessingReqSet"
• 3	.ReqSetListPage1.ReqSetList(1).Milestone	equal to	@(Pega-RULES:Utilities).getLocalizedT
• 4	.ReqSetListPage1.ReqSetList(1).TargetDeadLineDate	equal to	@toDate(@addDays(@CurrentDateTim
• 5	.ReqSetListPage1.ReqSetList(2).ReqSetClass	equal to	"PegaFS-Work-NewAccount"
• 6	.ReqSetListPage1.ReqSetList(2).ReqSetName	equal to	"MortgageUnderwritingReqSet"
• 7	.ReqSetListPage1.ReqSetList(2).Milestone	equal to	@(Pega-RULES:Utilities).getLocalizedT
• 8	.ReqSetListPage1.ReqSetList(2).TargetDeadLineDate	equal to	@toDate(@addDays(@CurrentDateTim
• 9	.ReqSetListPage1.ReqSetList(3).ReqSetClass	equal to	"PegaFS-Work-NewAccount"
• 10	.ReqSetListPage1.ReqSetList(3).ReqSetName	equal to	"MortgageClosingReqSet"
• 11	.ReqSetListPage1.ReqSetList(3).Milestone	equal to	@(Pega-RULES:Utilities).getLocalizedT
• 12	.ReqSetListPage1.ReqSetList(3).TargetDeadLineDate	equal to	@toDate(@addDays(@CurrentDateTim

Call superclass data transform

The model above creates a ReqSetListPage1 page of the Data-RequirementSetList class that has three entries:

- The MortgageProcessingReqSet rule is processed and applied to the milestone/stage processing with a target completion date of 30 days in the future.

RequirementSets: Mortgage Processing Requirement Set [Available]
CL: PegaFS-Work-NewAccount **ID:** MortgageProcessingReqSet **RS:** PegaFSSample:08-02-01

[Save as](#) [Actions](#) [Private edit](#) [X](#)

[Requirement Rules](#) [Parameters](#) [EForm Prep](#) [History](#)

Applies when [?](#)

Allow this requirement set to be manually added to work in process

LABEL	CLASS NAME	REQUIREMENT RULE NAME*	APPLIES WHEN	ROUTING ACTIVITY	OVERALL SLA ACTIVITY	STALE DATE SLA ACTIVITY
<input type="checkbox"/>	PegaReq-Data-ReqVerify-	Personal Identification	<input type="text"/> ?			
<input type="checkbox"/>	PegaReq-Data-ReqVerify-	Mortgage Application	<input type="text"/> ?			

[+](#)

Satisfaction Logic

ConditionalSuitabilityFilter

- The MortgageUnderwritingReqSet rule is processed and applied to the Underwrite Loan milestone/stage with a target completion date of 30 days in the future.

RequirementSets: MortgageUnderwritingReqSet [Available]
CL: PegaFS-Work-NewAccount **ID:** MortgageUnderwritingReqSet **RS:** PegaFSSample:08-02-01

[Save as](#) [Actions](#) [Private edit](#) [X](#)

[Requirement Rules](#) [Parameters](#) [EForm Prep](#) [History](#)

Applies when [?](#)

Allow this requirement set to be manually added to work in process

LABEL	CLASS NAME	REQUIREMENT RULE NAME*	APPLIES WHEN	ROUTING ACTIVITY	OVERALL SLA ACTIVITY	STALE DATE SLA ACTIVITY
<input type="checkbox"/>	PegaReq-Data-ReqVerify-	Credit Report	<input type="text"/> ?			
<input type="checkbox"/>	PegaReq-Data-ReqVerify-	Mortgage Income	<input type="text"/> ?			

[+](#)

Satisfaction Logic

ConditionalSuitabilityFilter

- The MortgageClosingReqSet rule is processed and applied to the Close Loan milestone/stage with a target completion date of 30 days in the future.

LABEL	CLASS NAME	REQUIREMENT RULE NAME*	APPLIES WHEN	ROUTING ACTIVITY	OVERALL SLA ACTIVITY	STALE DATE SLA ACTIVITY
	PegaReq-Data-ReqVerify-	MortgageClosingAgentContact				
	PegaReq-Data-ReqVerify-	MortgageHUD_1				

Applies when:

Allow this requirement set to be manually added to work in process

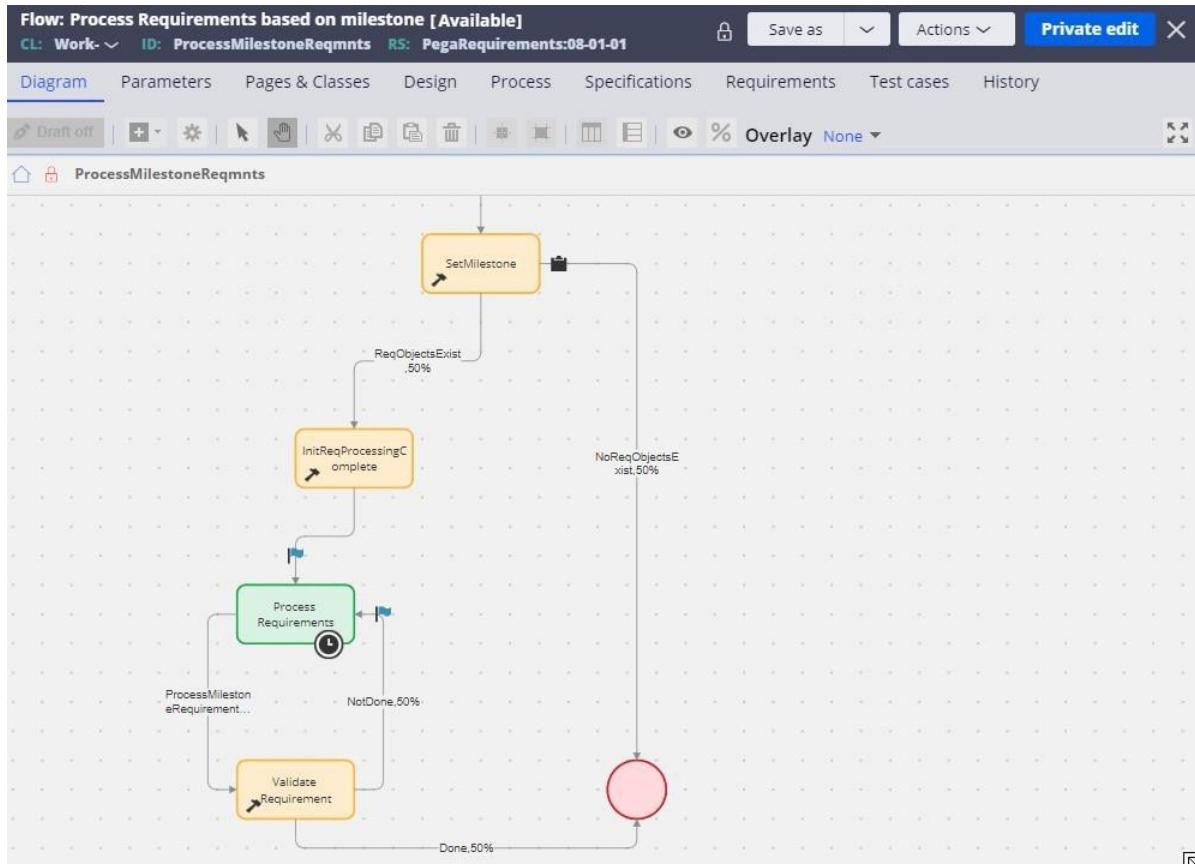
Satisfaction Logic:

ConditionalSuitabilityFilter:

Configuring stage processing

To set a stage point in the parent case processing flow, call the application.

Work-.ProcessMilestoneReqmnts subflow (or extend the flow to meet your own needs). In the following Mortgage processing flow, this subflow is called three times, once for each stage in the process. It is called with a parameter that indicates which stage should be processed.



Runtime results

At runtime, the FNMAConforming30YearFixedRateMortgage product described above is selected, and the mortgage property information is entered as shown below.

When the Requirement Sets for the product are processed, the necessary Requirements are created.

As specified by the product model, the system:

- Creates the requirements as listed in the MortgageProcessingReqSet, and applies them to the Process Loan stage. Note that because the mortgage property is in FL, the When rule PropertyIsInAZ is not True, and so the Proof of Residency requirement is not created for this parent case.
- Creates the requirements as listed in the MortgageUnderwritingReqSet, and applies them to the Underwrite Loan stage.
- Creates the requirements as listed in the MortgageClosingReqSet, and applies them to the Close Loan stage.

Passing data between a requirement and its parent case

The Requirements functionality implements the passing of data between the parent case and its requirements at two points in requirement processing.

When a requirement is instantiated, data can be passed from the parent case to the Requirement object to initialize it. If the data is used to validate individual documents, the Requirement rule can name an activity that is run to distribute the data appropriately among its document entries after it is passed to the requirement.

When a requirement is validated, data can be passed from the Requirement object to the parent case to update it with information gathered during the validation process. If the data is gathered from individual documents, the Requirement rule can name an activity to collect the data appropriately from its document entries prior to passing it back to the parent case.

There are two methods of passing data between the parent case and the Requirement object.

For passing simple scalar values between the objects, you can use the Parameter tabs of the Requirement rule and the Requirement Set rule to specify the data that is passed.

Advantages:

- Does not require that the Requirement rule know anything about the parent case internal data structure, or that the parent case know anything about the internal data structure of the requirement. This is useful if the requirement is used with parent cases of different work types.
- Minimizes the degree to which Requirement rules must be kept *in sync* with changes in the parent case structure, and vice versa.

Disadvantages:

- As implemented, the approach is limited to scalar values. It cannot be used to pass pages and lists.
- If many scalar values are being passed, configuration can be somewhat cumbersome.

- If the passed data is used by individual document validations in the requirement, an input activity to distribute the data to the documents (at initialization time) and an output activity to collect data from the documents (at validation time) must be defined for the Requirement rule. This is because the parameters always pass data at the requirement level, not at the individual document level.

For passing data such as page data or lists, or to process more complicated initializations or updates, the requirement Input and Output activity can open and work with the parent case directly.

Advantages:

- This approach can handle arbitrarily complex updates.
- Can be configured completely in the Requirement rule Input and Output activities, including the passing of data to and from individual requirement documents as needed.

Disadvantages:

- Requires that the Requirement rule know about the parent case internal data structure, and that the parent case know about the internal data structure of the requirement. This may be problematic if the requirement is used with parent cases of different work types.
 - You must pay more attention when making changes to the parent case structure to ensure that Requirement rules are kept *in sync* with changes in the parent case structure.
-
- **Passing scalar data using parameters**
 - **Passing data without parameters**

Passing scalar data using parameters

In this approach, you use the **Parameter** tabs of the Requirement rule and the Requirement Set rule to specify the data that is passed between the parent case and the Requirement object.

Configuration

The parameter page of the Requirement Set rule, which is normally defined on the parent case class, defines the properties in the parent case that are output to or received from the Requirement object. The following figures show both tabs of the Requirement Set MortgageUnderwritingReqSet rule. The **Requirement Rules** tab shows that two requirements are created when this Requirement Set is processed. The **Parameters** tab shows that two parent case properties are mapped to or from the MortgageCreditReport requirement:

The parent case .pyWorkParty (Applicant).TaxIDNbr property is mapped out from the parent case to the SocialSecNum parameter when the requirement is created.

The Credit Score parameter is mapped in to the parent case property .CreditScore when the requirement is validated.

The screenshot shows the Requirements Sets configuration page for 'MortgageUnderwritingReqSet'. The top navigation bar includes 'RequirementSets: MortgageUnderwritingReqSet [Available]', 'CL: PegaFS-Work-NewAccount', 'ID: MortgageUnderwritingReqSet', 'RS: PegaFSSample:08-02-01', 'Save as', 'Actions', and 'Private edit'.

The main area displays the 'Requirement Rules' tab. It shows two requirement rules:

LABEL	CLASS NAME	REQUIREMENT RULE NAME *	APPLIES WHEN	ROUTING ACTIVITY	OVERALL SLA ACTIVITY	STALE DATE SLA ACTIVITY
	PegaReq-Data-ReqVerify-	Credit Report	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	PegaReq-Data-ReqVerify-	Mortgage Income	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Below the table, there are sections for 'Satisfaction Logic' and 'ConditionalSuitabilityFilter'.

Requirement Rule Name*	Applies When	Routing Activity	Overall SLA Activity	Stale Date SLA Activity
Credit Report				
Mortgage Income				

In an analogous way, the parameter page of the Requirement rule defines the properties in the Requirement object that is output to or received from the parent case. Continuing the example below, the **Parameter** tab of the MortgageCreditReport Requirement rule shows that two Requirement object properties were mapped to or from the parent case:

The SocialSecNum parameter is mapped in to the .MortgageCreditReportSSN Requirement object property when the requirement is created.

The .MortgageCreditReportScore Requirement object property is mapped out from the Requirement object to the Credit Score parameter when the requirement is validated.

Be sure that you define all these properties in the same or an ancestor class of the class in which the Requirement rule is defined.

The DistributeCreditReportData Input activity runs after you load all the input parameters into the Requirement to move the input data to individual documents of the requirement as needed for validation.

The AssembleCreditReportData Output activity runs before all the output parameters are passed back to the parent case to move the validated data of individual documents to the requirement level to be passed back through parameters.

Requirements: Credit Report [Available]
CL: PegaReq-Data-ReqVerify- ID: MortgageCreditReport RS: PegaFSSample:08-02-01
This record has 1 info warning (including 1 unjustified) [View](#)

Name	In/Out	Property	Page?	Page class
SocialSecNum	in ▾	.MortgageCreditReportSSN	<input type="checkbox"/>	X
CreditScore	Out ▾	.MortgageCreditReportScore	<input type="checkbox"/>	X

Runtime results

As a result of the parameter settings in the Requirement Set and Requirement rules above, data is passed between the parent case and the Mortgage Credit Report Requirement object as follows. At the time the Requirement is created:

- The pyWorkParty (Applicant).TaxIDNbr parent case property is mapped out from the parent case to the SocialSecNum parameter.
- The SocialSecNum parameter is mapped in to the .MortgageCreditReportSSN Requirement object property.
- The DistributeCreditReportData Input activity moves the input data to individual documents of the requirement as needed for validation.

At the time the Requirement is validated:

- The AssembleCreditReportData Output activity moves the validated data of individual documents to the requirement level to be passed back through parameters.
- The .MortgageCreditReportScore Requirement object property is mapped out from the Requirement object to the Credit Score parameter.
- The Credit Score parameter is mapped in to the .CreditScore parent case property.

This results in the parent case receiving the credit score from the Requirement object at the time the requirement is validated.

Passing data without parameters

In this approach, the Input activity and Output activity entries on the Parameters tab of the Requirement rule are used to specify the data that is passed between the parent case and the Requirement object. No other entries should be made on the Parameters tab.

Configuration

At the time that the Requirement is created, the specified Input activity completely specifies the movement of data from the parent case to the overall Requirement object, as well as the movement of data from the parent case to individual requirement document validation pages.

At requirement validation time, the Output activity completely specifies the movement of data from the Requirement object.

Attachments in requirements

- [Attachment capabilities in Pega Platform](#)
- [Attachments using CMIS](#)

Attachment capabilities in Pega Platform

To use the Pegasystems attachment capabilities, open the application and ensure that Store in Pega database (default) is selected on the Integration & security tab.

Edit Application: Sample Application for PFFS 8.3

ID: PegaFSIF_Sample • 8 RS: PegaFSSample [Edit]

This record has 3 info warnings (including 3 unjustified) [Review/Edit]

Definition Cases & data Application wizard Documentation **Integration & security** History

Application security

Require password to update application [Update password](#)

Content management

Configure the storage location for case and pulse attachments. Optionally configure sources for case and pulse attachments.

 Content storage

Store in Pega database
 Store in CMIS system
 Store in web storage provider
 Store in repository

 Content sourcing

Source from CMIS systems
 Source from web storage providers
 Source from repositories

Agile Workbench integration

[Configure integration](#)

Attachments using CMIS

Content Management Interoperability Services (CMIS) is an open standard that allows different content management systems to interoperate over the Internet. Specifically, CMIS defines an abstraction layer for controlling diverse document management systems and repositories using web protocols.

CMIS provides a common data model covering categorized files and folders with generic properties that can be set or read. There is a set of services for adding and retrieving documents. There might be an access control system, a checkout and version control facility, and the ability to define generic relations. Three protocol bindings are defined: one using WSDL and SOAP, another using AtomPub, and a browser-friendly one using JSON. The model is based on common architectures of document management systems. To use a CMIS system ensure that **Store in CMIS repository** is selected on the **Integration & security** tab.

For the current implementation, the protocol using AtomPub has been leveraged. This requires the creation of an Atom Server rule in the application instance with the necessary details. A Dynamic System Setting called CMIS/URL in PegaFS has been set up to contain the URL of the CMIS server. This will need to be updated.

The screenshot shows the 'Edit Atom Server' form. At the top, it displays the ID 'vfwappudbi106' and the rule set 'PegaFS [Edit]'. Below the header are two tabs: 'Environment' (which is selected) and 'History'. The main area contains the following fields:

- Atom server URI ***: The value is '=D_CMISEndPointURL.pyAtomServerURI'.
- User ID**: The value is 'admin'.
- Password**: The value is '.....' (represented by dots).
- Use NTLMv1**: A checkbox is checked.

At the bottom of the form is a blue 'Test Connectivity' button.

A new Connect-CMIS rule was created and the new Atom Server instance is referenced in the rule. This rule is used in the Application rule form that helps the application instance interact with the CMIS server configured in the referenced rule forms.

Configure this rule on the **Integration and security** tab of the Application rule form in order to direct the application to connect to the specified server for managing content uploaded for work. Select the **Enable for attachments** check box to leverage the CMIS server capabilities.

The **CMIS Folder** field is the server folder where the uploaded files are stored. You must select this folder explicitly when you change the Connector (**Connector Name** field), and you should save the application rule form after you select the CMIS folder. If you select a connector and no folders are displayed, do the following: save the application rule form, browse for folders, select the folder, and then resave the application rule form. For consistency while uploading and searching for documents, create a folder on the server to hold your content.

Note: The configuration described above assumes that there is a working CMIS server in place. In this case, the server configured is an Alfresco instance.

- ⓘ However, that is not a requirement. You can use any CMIS-compatible server and configure it in the same way and the run-time behavior will not be affected.

- Configuration to support customer properties metadata
- Technical details for the manage requirement documents screen
- Configuring Pega Platform to act as a CMIS server

Configuration to support customer properties metadata

Once the CMIS configuration is in place, every document uploaded for work is uploaded to the CMIS system. The uploaded document includes the basic details of the document, such as content type, size, and name, as well as metadata. A document with just basic details would suffice if the document were for backend verification only. However, if the document is required for verification in the UI or can be reused later, then the basic details would not serve the purpose. Also, if no additional metadata is specified, a search on the instance would return all the documents uploaded to the server, which may not be the expected behavior.

The addition of metadata while uploading a document helps differentiate it from other documents based on the metadata configured. This also helps in searching for specific documents rather than querying the whole list.

For the current implementation, Document Type and Customer ID have been added as the custom metadata properties in addition to some Pega-provided properties. With the addition of these properties, you need to configure the CMIS server.

The sample configuration for *Alfresco* is as follows:

Create an XML file in the following location: <Alfresco installed drive>\Alfresco\tomcat\shared\classes\alfresco\extension.

Use the name pega-model.xml with the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Definition of new Model -->
<model name="pega:pegaModel" xmlns="http://www.alfresco.org/model/dictionary/1.0">

    <!-- Optional meta-data about the model -->
    <description>Pega Model</description>
    <author>Praneeth</author>
    <version>1.0</version>

    <!-- Imports are required to allow references to definitions in other models -->
    <imports>
        <!-- Import Alfresco Dictionary Definitions -->
        <import uri="http://www.alfresco.org/model/dictionary/1.0" prefix="d" />
        <!-- Import Alfresco Content Domain Model Definitions -->
        <import uri="http://www.alfresco.org/model/content/1.0" prefix="cm" />
    </imports>

    <!-- Introduction of new namespaces defined by this model -->
    <namespaces>
        <namespace uri="http://www.pega.com/model/content/1.0" prefix="pega" />
    </namespaces>

    <types>
        <type name="pega:workObjectAttachment">
            <title>work object attachment</title>
            <parent>cm:content</parent>
            <properties>
                <property name="pega:pxAttachKey">
                    <type>d:text</type>
                    <mandatory>true</mandatory>
                </property>
                <property name="pega:pxRefObjectKey">
                    <type>d:text</type>
                    <mandatory>true</mandatory>
                </property>
                <property name="pega:CustomerID">
                    <type>d:text</type>
                    <mandatory>false</mandatory>
                </property>
                <property name="pega:DocumentType">
                    <type>d:text</type>
                    <mandatory>false</mandatory>
                </property>
            </properties>
        </type>
    </types>
</model>
```

You can copy the XML documents provided in this section.

The pega:workObjectAttachment is the type of object uploaded in our application that will be stored in CMIS. The pega:pxAttachKey, pega:pxRefObjectKey, pega:DocumentType, and pega:CustomerID are the properties that are uploaded to the CMIS while uploading the document. These properties can be further used to query the CMIS to search for the uploaded documents.

Using the XML below, create an XML file that references the pega-model.xml in the following directory: <Alfresco installed drive>\Alfresco\tomcat\shared\classes\alfresco\extension

The name should contain “-context” because the application configuration checks for any additional configuration in the “*-context.xml” files. As a result, the configuration described above is picked up from this file.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE beans PUBLIC '-//SPRING//DTD BEAN//EN' 'http://www.springframework.org/dtd/spring-beans.dtd'>
<beans>
<!-- Registration of new models -->
<bean id="extension.dictionaryBootstrap" parent="dictionaryModelBootstrap" depends-on="dictionaryBootstrap">
<property name="models">
<list>
<value>alfresco/extension/pega-model.xml</value>
</list>
</property>
</bean>
</beans>
```

If the web-client-config-custom.xml file does not exist in the following folder, create it in the following directory, and then update it with the XML shown in the subsequent figure:

<Alfresco installed drive>\Alfresco\tomcat\shared\classes\alfresco\extension

```
<alfresco-config>
<config evaluator="node-type" condition="pega:workObjectAttachment">
<property-sheet>
<show-property name="name" show-in-edit-mode="false" />
<show-property name="mimetype" display-label-id="mimetype"
    converter="org.alfresco.faces.MimeTypeConverter"
    show-in-edit-mode="false" />
<show-property name="title" show-in-edit-mode="false" />
<show-property name="description" show-in-edit-mode="false" />
<show-property name="size" display-label-id="size"
    converter="org.alfresco.faces.ByteSizeConverter"
    show-in-edit-mode="false" />
<show-property name="pega:pxAttachKey" display-label-id="Attach Key" show-in-edit-mode="false" />
<show-property name="pega:pxRefObjectKey" display-label-id="Reference Object Key" show-in-edit-mode="false" />
<show-property name="pega:DocumentType" display-label-id="Document Type" show-in-edit-mode="false" />
<show-property name="pega:CustomerID" display-label-id="Customer ID" show-in-edit-mode="false" />
</property-sheet>
</config>

<config evaluator="string-compare" condition="Content Wizards">
<content-types>
<type name="pega:workObjectAttachment" display-label="Pega WO" display-label-id="pwo" />
</content-types>
</config>
</alfresco-config>
```

The configuration above is specific to Alfresco. If you specify a different CMIS server, then the directories and configuration may change based on the configuration of the server.

Also, you might need to change the names for different CMIS servers. In addition to configuring the files as specified, you need to restart the CMIS server for the changes to take effect. Also, you need to save the Application rule form after you restart the CMIS server.

Technical details for the manage requirement documents screen

The Manage Documents screen is displayed using the ManageReqDocLinks section and its referencing flow action.

Section: Manage Requirement Documents Links [Available]

CL: PegaReq-Data-ReqDocEntry ID: ManageReqDocLinks RS: PegaRequirements:08-02-01

Design Settings Parameters Pages & Classes HTML Specifications History

Save as Actions Private edit >

Table - 1
Linked document
Table [DocumentActivityStatusList of Class History-DocumentActivity]

DocName	DocType	DocumentActivityDate...	pyMessageKey	UnlinkButtonLabel
---------	---------	-------------------------	--------------	-------------------

Dynamic Layout (Inline-middle) - 2

MatchingDocuments

Table - 3
Matching documents
Table [lcmQueryResults.pyObjects.pyObjects of Class CMIS-CmisObjectType]

Document name	Type	Version	Date	LinkButtonLabel
---------------	------	---------	------	-----------------

Dynamic Layout (Default) - 4

UploadAndLinkButtonLabel

Table - 5

Flow Action: Manage Requirement Documents [Available]
CL: PegaReq-Data-ReqDocEntry **ID:** ManageReqDocLinks **RS:** PegaRequirements:08-01-01

Layout Validation Action Help setup Security HTML Pages & Classes Specifications History

Section configuration

Page context
Use current page context

Applies to
PegaReq-Data-ReqDocEntry

Section
ManageReqDocLinks Edit parameters

Refresh settings

Refresh condition

The Upload and Link screen is displayed using the AttachFileBrowseMultiOptionCMIS section and its referencing flow action.

Section: Browse upload for Multi File DD [Available, Internal]
CL: Data-WorkAttach-File **ID:** AttachFileBrowseMultiOptionsCMIS **RS:** PegaRequirements:08-02-01

Design Settings Parameters Pages & Classes HTML Specifications History

Upload and link to document

File Name
HIDDEN CONTENT

Document ID
DocumentID

Comment category*
Document Reviewed

Comments

File
Choose File No file chosen

Flow Action: Attach File Browse MultiOptions CMIS [Available]
CL: Work- **ID:** AttachFileBrowseMultiOptionsCMIS **RS:** PegaRequirements:08-01-01

Save as Actions Private edit X

Layout Validation Action Help setup Security HTML Pages & Classes Specifications History

Section configuration

Page context
Use clipboard page ▾

Page *
pyAttachmentPage

Applies to
Work-

Section
AttachFileBrowseMultiOptionsCMIS Edit parameters

Refresh settings

Refresh condition

The uploading and linking of the document is handled by the UploadAndLinkDocument activity.

Activity: Upload And Link Document [Available]
CL: Work- **ID:** UploadAndLinkDocument **RS:** PegaFSRequirements:08-01-01

Save as Actions Private edit

This record has 2 justified warnings View

Steps Parameters Pages & Classes Security Test cases Specifications History

Label	Method	Step page	Description	Jump	Remove
1.	Loop When > Page-Clear-Messages	Param.AttachmentPageNa	Clear all messages on the step page	Jump	Remove
2.	Loop When > Property-Set	Param.AttachmentPageNa	Set parameters required for processing file.	Jump	Remove
3.	Loop When > Property-Set	Param.AttachmentPageNa	Set property values	Jump	Remove
4.	Loop When > Page-Change-Class	Param.AttachmentPageNa	Change the page class	Jump	Remove
5.	// Loop When > Property-Set	Param.AttachmentPageNa	Iterate Through page	Jump	Remove
6.	// Loop When > Property-Set	Param.AttachmentPageNa	Iterate Through page	Jump	Remove
7.	Loop When > Property-Set	Param.AttachmentPageNa	Set the pyNote	Jump	Remove
8.	Loop When > Apply-DataTransform	Param.AttachmentPageNa	Run ManageAssociations DT	Jump	Remove
9.	PRPC Loop When > Property-Set	pyWorkPage.ReqDocs	Set version id when the connector is PRPC	Jump	Remove
10.	Loop When > Call pzSaveAllAttachments	pyWorkPage	Call activity to save attachment	Jump	Remove
11.	Loop When > Property-Set	Param.AttachmentPageNa	Set property values	Jump	Remove
12.	Loop When > Call SelectAttachment	pyWorkPage	Retrieve the latest uploaded documents.	Jump	Remove
13.	Loop When > Property-Set	ecmQueryResults.pyObject	Call activity for linking the uploaded document.	Jump	Remove
1.	Loop When > Property-Set	Param.AttachmentPageNa	Set default comments when in good order	Jump	Remove

The document upload is handled by Step 7 of the activity.

As described earlier, the document upload includes metadata. This is configured in the pySaveFileContent activity. Step 11 of the activity has been configured to add the required metadata, which is uploaded as properties associated with the document. Note that the property IDs or names in this example are specific to an Alfresco server. If you use a different CMIS server, then modify the property names per the configuration of the CMIS server.

*PropertiesName	*PropertiesValue
.pyProperties.pyPropertyId(1).pyValue	"D:pega:workObjectAttachment"
.pyProperties.pyPropertyString(2).pyValue	"pega:pxAttachKey"
.pyProperties.pyPropertyString(2).pyValue	Primary.pxAttachKey
.pyProperties.pyPropertyString(3).pyValue	"pega:pxRefObjectKey"
.pyProperties.pyPropertyString(3).pyValue	Primary.pxRefObjectKey
.pyProperties.pyPropertyString(4).pyValue	"pega:DocumentAssociationInformation"
.pyProperties.pyPropertyString(4).pyValue	CMISParamPage.DocumentAssociationInformation
.pyProperties.pyPropertyString(5).pyValue	"pega:DocumentType"
.pyProperties.pyPropertyString(5).pyValue	pyWorkPage.TempSelectedDoc

The basic CMIS upload functionality has been extended for uploading a new version of the document using the InvokeCMISConnector activity. Step 11 has been updated for the change; this works for any CMIS server. No specific updates are needed at this level for uploading a new version of the document.

Activity: Invoke CMIS Connector [Available]

Save as Actions Private edit

8.	Loop When > Obj-Open	TempSOAPServer	Open SOAP server instance	Jump
9.	Loop When > Java		Initialize the OpenCMIS session object	Jump
10.	PERFO Loop When > Java	pyObjName	Invoke the getRepositoryInfo service	Jump
11.	Loop When > Java	pyObjName	Invoke the query service	Jump

Method Parameters

```

1 com.pega.apache.chemistry.opencmis.client.api.Session session =
2 (com.pega.apache.chemistry.opencmis.client.api.Session)sessionObj;
3
4 // get the SQL statement from the request data page
5 ClipboardPage requestDataPage = myStepPage.getPage("pyrequestData");
6 String statement = requestDataPage.getString("pyStatement");
7 /*statement="SELECT cmis:objectId, cmis:name, cmis:versionLabel, cmis:creationDate, cmis:contentStreamMimeType, pega:pxAttachKey FROM
8 pega:workObjectAttachment WHERE IN_FOLDER('workspace://SpacesStore/384810c5-7c68-4274-bc2b-75643754d6c1') AND (pega:DocumentType LIKE 'Passport') AND (pega:DocumentAssociationInformation IN ('Individual:0000333333;'))";*/
9 if (statement.length() == 0)
10 throw new ConnectorException("pyStatement property of class CMIS-Query cannot be blank");
11
12 //resolve property references
13 int startInd = statement.indexOf("{");
14 while(startInd>0){
15     int pos = statement.indexOf("}", startInd);
16     String propRef = statement.substring(startInd+1, pos);
17     String targetProp = ((ClipboardProperty)tools.getProperty(propRef)).getStringValue();
18     statement= statement.substring(0, startInd)+ targetProp + statement.substring(pos+1);
19     startInd = statement.indexOf("{");
20 }
21
22 if (oLog.isDebugEnabled())
23     oLog.debug("Executing CMIS query for statement: " + statement);
24

```

All operations related to the CMIS use this activity. Be careful when modifying any of the functionality in the specified activity.

As shown earlier, the **Matching Document** section in the Manage Documents screen employs search on the CMIS server, based on the uploaded metadata. That is, it uses the uploaded document metadata to search for the document. This is done in the FindECMDocuments activity. Step 8 calls the PrepareCMISQuery data transform that sets the necessary details for searching. Additional metadata configured could be added to the query if you want to use that metadata to search for the data transform.

Activity: FindECMDocuments [Available]

Save as Actions Private edit

4.	Loop When >	Page-New	CMISRequestData	Set up request	Jump
5.	Loop When >	Page-New	CMISResponseData	Set up return	Jump
6.	Loop When >	call pyGetDefaultEndpoint	ServicePage	Get the service	Jump
7.	// Loop When >	Property-Set	CMISRequestData	Fill the props	Jump
8.	Loop When >	Apply-DataTransform		Call data transform which sets the CMIS search que	Jump

Method Parameters

Name	Value
* DataTransform	PrepareCMISQuery
PassParameterPage	

9.	// Loop When >	Property-Set	CMISRequestData	Add the folder condition	Jump
10.	// Loop When >			If EndPointName has a value then we will not default	Jump
11.	Loop When >	call pyCMISConnector	ServicePage	call to create	Jump

Data Transform: Prepare CMIS Query [Available]

CL: Work- ID: PrepareCMISQuery RS: PegaRequirements:08-01-01

Save as Actions Private edit

Definition Parameters Pages & Classes Test cases Specifications History

Action	Target	Relation	Source
Comment	CMISRequestData	equal to	100
Set	.pyMaxItems	equal to	CMISParamPage.IsAnyVersionSearch
Set	.pySearchAllVersions	equal to	@(Pega-RULES:Utilities.escapeSpe
Set	Param.DocType	equal to	@(Pega-WB:ReportContentBuilder)
Set	Param.DocType	equal to	"SELECT cmis:objectId, cmis:name,
When	Param.DocType=""		
Set	.pyStatement	equal to	CMISRequestData.pyStatement+@
Set	Param.TempOwnerValue	equal to	@PegaRequirements:PegaRequirement

Once the document has been uploaded, a provision automatically uses the LinkSelectedDocument activity to link the uploaded document to the selected Document Requirement. That activity, in turn, calls the LinkSelectedDocumentToCase data transform, which sets the necessary properties required for linking the document to the requirement. The document commentary is also updated as a part of this data transform. The same activity is also used when you click on the **Link** button, thus maintaining a consistent process.

Activity: Link Selected Document [Available]

CL: Data- ID: LinkSelectedDocument RS: PegaRequirements:08-01-01

This record has 1 justified warning [View](#)

Steps Parameters Pages & Classes Security Test cases Specifications History

Label	Method	Step page	Description	Action
1.	Loop When > Page-Set-Messages		Set error messages on page if needed.	Jump Delete
2.	Loop When > Property-Set	ecmQueryResults.pyObjec	Set the linked flag to no if any document already linked	Jump Delete
3.	Loop When > Property-Set		Set the reqdocs page's full name to a parameter for fut	Jump Delete
4.	Loop When > Page-Copy		Copy the Primary page into a temp page for reference	Jump Delete
5.	Loop When > Property-Set	param.TempReqDocPage	Loop through the requirement documents	Jump Delete
1.	Loop When > Apply-DataTransform	param.TempReqDocPage	Call data transform to link the selected document to	Jump Delete
2.	Loop When > Property-Set		Reset the parameter page value.	Jump Delete
6.	// Loop When > Apply-DataTransform		Apply a data transform	Jump Delete
7.	Loop When > Call UpdateStatus	pyWorkPage	Update status of the case.	Jump Delete
8.	Loop When > Obj-Refresh-And-Lock	pyWorkPage	Lock the work object	Jump Delete
9.	Loop When > Obj-Save	pyWorkPage	Save the work object	Jump Delete
10.	Loop When > Call CommitWithErrorHandler	pyWorkPage	Commit the changes explicitly.	Jump Delete

[Add a step](#) [Collapse all steps](#)

Data Transform: Link Selected Document To Case [Available]

CL: PegaReq-Data-ReqDocEntry ID: LinkSelectedDocumentToCase RS: PegaRequirements:08-01-01

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

Definition Parameters Pages & Classes Test cases Specifications History

Action	Target	Relation	Source	Action
For Each Page In	.DocumentActivityStatusList		<input type="checkbox"/> Also use each page as source context	Delete
• 1.1	.IsLatest	equal to	"false"	Select values + Edit Delete
• 2	Param.ResultCount	equal to	.DocumentActivityStatusList(<LAST>)	Select values + Edit Delete
• 3	.DocumentActivityStatusList(<APPEND>).pxObjClass	equal to	"History-DocumentActivity"	Select values + Edit Delete
• 4	.DocumentActivityStatusList(<APPEND>).DocObject	equal to	TempPrimaryPage.pyObjectID	Select values + Edit Delete
• 5	.DocumentActivityStatusList(<LAST>)			Delete
• 5.1	Apply Data Transfc			Edit Delete
• 6	Set .DocumentActivityStatusList(<LAST>).DocVersion	equal to	TempPrimaryPage.DocumentVersion	Select values + Edit Delete
• 7	Set .DocumentActivityStatusList(<LAST>).DocName	equal to	TempPrimaryPage.DocumentName	Select values + Edit Delete
• 8	Set .DocumentActivityStatusList(<LAST>).DocumentActi	equal to	CMISObjectPage.DocumentActivityCa	Select values + Edit Delete
• 9	When CMISObjectPage.IsVerificationReady=="true"			Delete
• 10	Set DocumentActivityStatusList(<LAST>).DocumentActi	equal to	"Document Reviewer"	Select values + Edit Delete

As described earlier, a linked document includes an **Unlink** button. Click it to unlink the document from the Document Requirement. This is done using the `unlinkDocument` activity.

Activity: Unlink Document [Available]

CL: History-DocumentActivity ID: UnlinkDocument RS: PegaRequirements:08-01-01

This record has 1 justified warning [View](#)

Steps Parameters Pages & Classes Security Test cases Specifications History

Label	Method	Step page	Description	Jump	Delete
1. <input type="text"/> Loop When > <input type="button" value="Property-Set"/>	<input type="button" value="Property-Set"/>	<input type="text"/>	Set the requirement document page name to a parameter	Jump	Delete
2. <input type="text"/> Loop When > <input type="button" value="Property-Set"/>	<input type="button" value="Property-Set"/>	<input type="text"/>	loop through the reqdocs page	Jump	Delete
1. <input type="text"/> Loop When > <input type="button" value="Property-Set"/>	<input type="button" value="Property-Set"/>	<input type="text"/> param.TempReqDocPage	Set necessary properties for unlinking document	Jump	Delete
2. <input type="text"/> Loop When > <input type="button" value="Property-Set"/>	<input type="button" value="Property-Set"/>	<input type="text"/>	Reset the param page name	Jump	Delete
3. // Loop When > <input type="button" value="Property-Set"/>	<input type="button" value="Property-Set"/>	<input type="text"/> pyWorkPage.ReqDocs	Set property values	Jump	Delete
4. <input type="text"/> Loop When > <input type="button" value="Property-Set"/>	<input type="button" value="Property-Set"/>	<input type="text"/> ecmQueryResults.pyObject	Set the linked flag to false	Jump	Delete
5. <input type="text"/> Loop When > <input type="button" value="Obj-Refresh-And-Lock"/>	<input type="button" value="Obj-Refresh-And-Lock"/>	<input type="text"/> pyWorkPage	Acquire lock on the object	Jump	Delete
6. <input type="text"/> Loop When > <input type="button" value="Obj-Save"/>	<input type="button" value="Obj-Save"/>	<input type="text"/> pyWorkPage	save the work object	Jump	Delete
7. <input type="text"/> Loop When > <input type="button" value="Call CommitWithErrorHandler"/>	<input type="button" value="Call CommitWithErrorHandler"/>	<input type="text"/> pyWorkPage	Commit the changes explicitly.	Jump	Delete

[+ Add a step](#) [Collapse all steps](#)

Configuring Pega Platform to act as a CMIS server

As discussed, the Manage Documents functionality leverages multiple features of Content.

Management Interoperability Services (CMIS) standards, which requires a CMIS server. But what if a CMIS server is not configured?

In that case, you can configure Pega Platform to act as a CMIS server. This requires that you create a Connect-CMIS rule with “PRPC” as the Atom Server. By default, Pega Platform does not include the capabilities that are provided by a CMIS server; therefore, you must create a dummy Connect-CMIS rule in the application named “PRPC”.

This record will be upgraded to a new version when saved.

Endpoint

Server type *

Atom

Server instance *

vfwlapudbi106

Connection

Timeout *

30000

Service

While uploading a document with another CMIS server in place, Pega Platform usually maintains a pointer to the CMIS object. If the CMIS server is Pega Platform, the page updates with metadata details that the CMIS-related activities have built, and the object is saved to the database table mapped to the Data-WorkAttach-File class. The way that the metadata is built has not changed.

However, the metadata page is now stored in the **BLOB** column of the Pega Platform table.

The ManageContentThroughPRPC activity (see the following screen) shows the changes that were made for uploading the attachment object to Pega Platform with the metadata built for CMIS. This activity is called in the InvokeCMISConnector activity. The flow branches out to this activity when the connector is Pega Platform; otherwise, it continues its normal protocol of connecting to the CMIS server.

Activity: Manage Content Through PRPC [Available]

CL: CMIS-ServiceData ID: ManageContentThroughPRPC RS: CMISPlus:08-01-01

This record has 2 justified warnings View

Steps Parameters Pages & Classes Security Test cases Specifications History

Label	Method	Step page	Description	Jump	Edit
1.	Loop When > Page-Remove	ecmQueryResults	Remove page(s)	Jump	Edit
2.	Loop When > Page-Copy	pyCMISConnectorPage	Copy the document details clipboard page to a tem	Jump	Edit
3.	Loop When > Property-Set		Set other properties required for saving the worko	Jump	Edit
4.	// Loop When > Obj-Save	pyNewFileAttachment	Save the document to the database.	Jump	Edit
5.	Loop When > Call CallObjSaveAndDelete	pyNewFileAttachment	Save the document to the database.	Jump	Edit
6.	Loop When > Obj-Refresh-And-Lock	pyNewFileAttachment	Refresh the attachment page and lock it bacuse th	Jump	Edit
7.	Loop When > Obj-Refresh-And-Lock	pyWorkPage	Refresh the workpage and lock it bacuse this is sav	Jump	Edit
8.	REQ Loop When > Obj-Browse		Find the latest uploaded document. Obj-Open is th	Jump	Edit
9.	SEARC Loop When > Obj-Browse		Find the latest uploaded document. Obj-Open is th	Jump	Edit
10.	// Loop When > Obj-Browse		Find the documents linked to the customer from th	Jump	Edit
11.	Loop When > Page-New	ecmQueryResults	Create a page for returning search results.	Jump	Edit
12.	Loop When >	CMISearchResults.pxResu	Loop through the search results.	Jump	Edit

The only other change for uploading a version document in Pega Platform is done in the pySaveFileContent activity. Step 14 has been added to update the version manually since Pega Platform does not have a mechanism to increment the version when uploading a document.

Activity: SaveFileContent [Available]

Save as Actions Private edit

13.	Loop	When >	Property-Set	CMISRequestData	Set request parameters	Jump	Remove
14.	Loop	When >	Property-Set	CMISRequestData	Set custom properties (if supported)	Jump	Remove
15.	Loop	When >	Property-Set	CMISRequestData	Set the Generated document name	Jump	Remove
16.	Loop	When >	Property-Set		Set property values	Jump	Remove
17.	Loop	When >	Property-Set	param.TempReqDocPage	Set property values	Jump	Remove
1.	Loop	When >	Property-Set	param.TempReqDocPage	Set property values	Jump	Remove

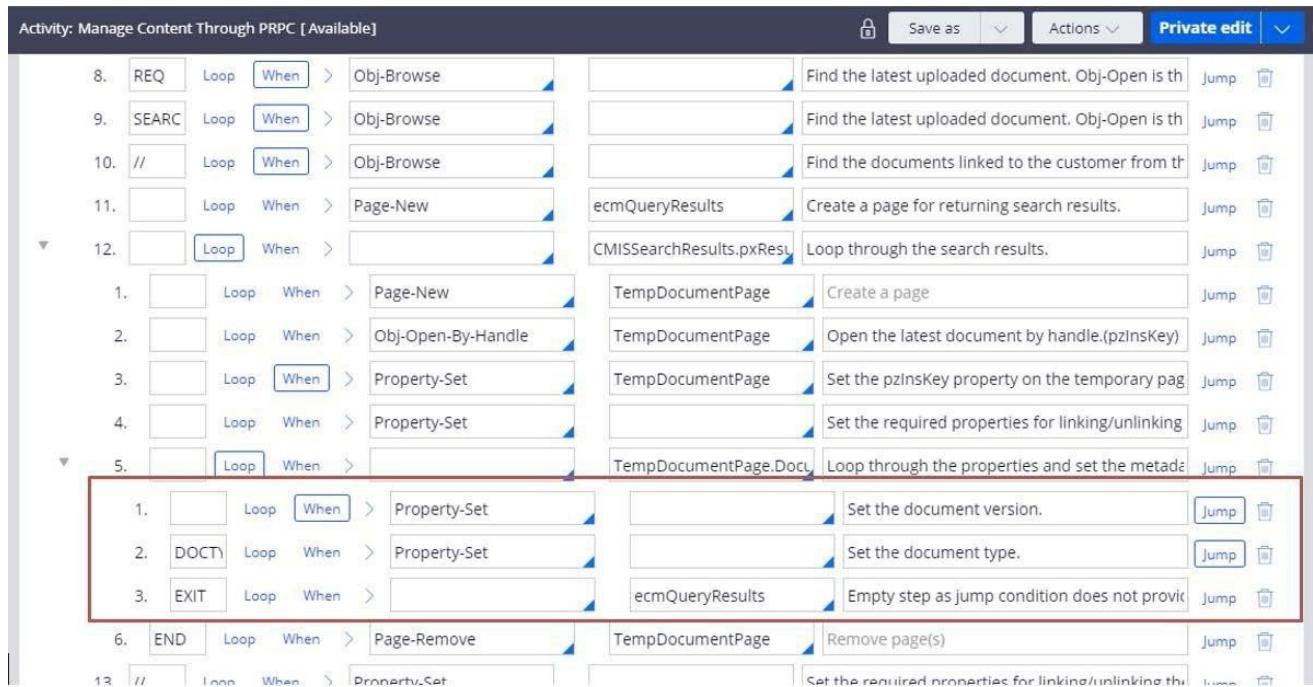
Method Parameters

*PropertiesName	*PropertiesValue
CMISRequestData.pyProperties.py\$	"cmis:VersionLabel"
CMISRequestData.pyProperties.py\$	@if(.DocumentActivityStatusList(<LAST>).DocVersion=="",@(Pega-RULES:String).toDecimal(1.0),@(Pega-RULES:String).toDecimal(1.0))
Param.IsIncremented	true

(+)

2.	Loop	When >	Property-Set		Set property values	Jump	Remove
18.	CMIS	Loop When >	Property-Set	CMISRequestData	Modify custom properties if no namespaces allowed	Jump	Remove
19.	Loop	When >	call SetCMISDocMetadataF.		Set CMIS Doc Metadata Fxt	Jump	Remove

The metadata update is carried out in the pySaveFileContent activity according to the Requirement. All the updates are persisted in the database. During the retrieval of the metadata, it is mapped to specific properties as required by CMIS for proper display. The ManageContentThroughPRPC activity maps the metadata retrieval as shown in the following steps:



After the document has been uploaded, the latest uploaded object's details are displayed and then mapped to the required properties for display in the UI (as shown in the screenshot above). For anyone extending the functionality, you need to update this step.

Search for the label provided during the metadata upload, and then set the value to the required property. Note that search uses the same functionality.

The DownloadContentStream activity is updated to add the steps needed for downloading and opening a Pega Platform document.

Activity: send content of CMIS document to browser [Available]

CL: @baseclass ID: DownloadContentStream RS: CMISPlus:08-01-01

Save as Actions Private edit

This record has 2 unreviewed warnings View

Steps	Parameters	Pages & Classes	Security	Test cases	Specifications	History
1.	Loop When > Page-New	ServicePage	Create Service Page			Jump
2.	Loop When > Page-New	CMISrequestData	Create Request Page			Jump
3.	Loop When > Property-Set	CMISrequestData	Populate Request Page			Jump
4.	Loop When > Call pyGetContentStream	ServicePage	Obtain Content of CMIS Document			Jump
5.	Loop When > Property-Set	ECMFileContent	Move stuff to file			Jump
6.	Loop When > java	CMISresponseData	Send Document Content to Browser from File			Jump
7.	WEBST Loop When > Obj-Open-By-Handle	pyAttachmentPage	Open record by handle			Jump
8.	Loop When > Call pyDownloadAttachmentContent	pyAttachmentPage	Call activity pyDownloadAttachmentContent			Jump
9.	Loop When > Page-Remove	pyAttachmentPage	Remove page(s)			Jump
10.	Loop When > Page-Remove	LinkPage	Remove page(s)			Jump
11.	REPOS Loop When > Obj-Open-By-Handle	pyAttachmentPage	Open record by handle			Jump
12.	Loop When > Call pyDownloadAttachmentContent	pyAttachmentPage	Call activity pyDownloadAttachmentContent			Jump
13.	Loop When > Page-Remove	pyAttachmentPage	Remove page(s)			Jump

In addition, a new SetCMISSearchParams data transform is added to set the required metadata on a temporary page for use by the upload, link, and search functionalities of CMIS. If you extend the functionality, update this data transform to set the required metadata on this page and use it later. This is called by the SelectAttachment activity, which is called when you click **Manage Documents** on the Requirements page.

Requirement rule form help

- **About requirement rules**
- **Rule resolution**
- **Requirements - Completing the Requirement tab**
- **Requirements - Completing the Parameters tab**
- **Requirements - Completing the Subject Tags tab**

About requirement rules

Purpose

Requirement rules (Rule-Requirement) control the attributes and behavior of the Requirement objects that are created at runtime and linked to a stage in the process loan flow of a parent case. The parent case can be any work object (that is, any concrete class derived from Work-).

Requirement Rules are referenced in RequirementSet rules.

Access

Use the Application Explorer to access activity rules that apply to the work types in your application. Use the Records Explorer to list all Requirement rules available to you.

Category

Requirement rules are part of the Requirement category. They are instances of the Rule- Requirement class.

Completing the new or Save As form

A requirement rule has two key parts.

Field	Description
Label	Enter a name for this Requirement, starting with a letter and using only alphanumeric and dash characters. Choose a name that is unique within the Applies To class. The name must be a valid Java identifier. The length of the class name plus the length of the Requirement Name cannot exceed 128 bytes.

Field	Description
Applies To	<p>Identify the class to which this rule applies.</p> <p>Typically, the Requirement rule is saved in a Data-class that also contains the Verification Section rules, Validation rules, Processing Flow rules as well as the various properties that must be passed to the parent case.</p> <p>The classes that you enter here depend on the ruleset you select. On the Restrictions tab of the Class form, a class rule may limit rules applying to that class to belong to one of an explicit list of rulesets.</p>

Rule resolution

Full rule resolution applies to Requirement rules. When searching for Requirement rules, the system does the following:

- Filters candidate rules based on a requestor's list of rulesets and versions
- Searches through ancestor classes in the class hierarchy for candidates when no matching rule is found in the starting class.

Requirements - Completing the Requirement tab

This tab is divided into two parts. The first part defines the rules pertaining to the overall Requirement work object. The second part defines the rules pertaining to the Document Requirement instances. The topics below provide a more detailed explanation:

Part 1: Defines the rules of the Overall Requirement work object

Field	Description
Verification Section	TheRule-HTML-Section rule used at runtime to display the section rule that is used to verify the overall validation for the Requirement.
Validation	TheRule-Obj-Validate rule used at runtime to perform the overall validation of the Requirement.
Processing Flow Class	If theRequirement is a flow-based Requirement, this field defines the class in which the flow resides.
Processing Flow Name	If the Requirement is a flow-based Requirement, this field defines the flow name.
Days Until Stale	Once the requirement is satisfied, this optional parameter indicates the number of days that the satisfied state is still considered valid. Used to calculate a Stale Date for the requirement; that is, the date when it should no longer be considered valid.
Processing Instructions	Instructions to be displayed to the operator processing the requirement at runtime.

Part 2: Defines the rules of the Requirement Document instances (there could be multiple)

Field	Description

Document ID	The DocumentID from the Document Library that is to be used to generate this line item. Pressing the Down Arrow key (smart prompt) lists the various Document Requirements from the Document Library. The class of these Document Library instances is picked up at runtime from the Declare_Doc_Settings Rule-DeclarePages rule.
Description	The name used for display purposes on various harnesses – more user-friendly than DocumentID.
Completion	Indicates whether the runtime validation of this document entry is successful.
Required	Mandatory for the satisfaction of this requirement.
Applies When	If not blank, names a When rule that is applied to determine if the document type is applicable to a particular parent case at runtime based on properties in the parent case. The When rule typically is defined on the same class or an ancestor class of the parent case class.
Attachment Category	If not blank, determines the Attachment Category to which the generated document form is attached on the Requirement work object at runtime.

Separate Doc for Each Party	If checked, indicates that this Document Requirement is created as a separate instance for each of the work parties on the Requirement work object. For example, if the Document Requirement is PayStub, selecting this check box creates a separate instance of the PayStub Document Requirement for each of the work parties associated with the Requirement work object.
Party Applies When	This field becomes visible only when the check box is selected for the Separate Doc for Each Party field. It points to the Rule-Obj-When rule used to evaluate whether the Document Requirement is applicable to each party based on certain conditions. For instance, if you select Separate Doc for Each Party, then the Applies When might indicate that the Document Requirement is applicable only when the party (that is workParty) is from the State of Massachusetts.
Reuse Existing	If selected, looks for existing Document Item instances to re-use instead of creating a new one. The criteria used to look for existing document instances is a combination of DocumentID, PartyID, and Detailed Description on the Document instances.
Verification Section	The Rule-HTML-Section rule used at runtime to display the section rule, which is used to verify the Document Requirement.

Validation	The Rule-Obj-Validate rule used at runtime to validate the document entry.
Valid Days	Number of days the document is considered valid before it expires.
Image Sys Params	For implementation use in integration with image repository systems.

Requirements - Completing the Parameters tab

This tab contains the fields described in this table.

Field	Description
Input Activity	<p>The name of an activity to run after you have loaded all the input parameters into the requirement. The activity moves the input data to the validation pages of individual documents of the requirement as needed for validation.</p> <p>Definethis activity on the same class or an ancestor class of the class on which the Requirement rule is defined.</p> <p>For an example, see the MortgageIncome requirement or the CreditScore requirement.</p>
Output Activity	<p>The name of an activity to run before the output parameters are passed back to the parent case. This activity moves the validated data of individual document validation pages</p>

Field	Description
	<p>to the requirement level to be passed back through parameters.</p> <p>Define this activity on the same class or an ancestor class of the class on which the Requirement rule is defined.</p> <p>For an example, see the MortgageIncome requirement or the CreditScore requirement.</p>
Name	The name of the Parameter used to exchange data between the Requirement work object and the parent case.
Description	The description of the parameter used to exchange data between the Requirement work object and the parent case.
In/Out/Both	<p>Accepts parameters IN / OUT / BOTH.</p> <p>IN -Indicates that the parameter is passed IN to the requirement object defined by this Requirement rule from the parent case at the time the requirement is created.</p> <p>OUT- Indicates that the parameter is passed OUT to the parent case from the requirement object defined by this Requirement rule when the requirement is validated.</p> <p>BOTH - Indicates that the parameter is passed IN to the requirement object defined by this Requirement rule from the parent</p>

Field	Description
	case at the time the requirement is created, and also passed OUT to the parent case from the requirement object defined by this Requirement rule when the requirement is validated.
Property	The name of the property that contains the data to be mapped between the parent case and the parameter page. This property must be a single value scalar property that is present in the Applies To class of the Requirement rule definition or an ancestor class.

Requirements - Completing the Subject Tags tab

This tab contains the following fields:

Field	Description
Name	Each entry is a text tag that identifies or categorizes the requirement in some way. Multiple tags are allowed. The tags are intended for use in creating reports or filtering lists of requirements by category.

Requirement set rule form help

- [About Requirement Set rules](#)
- [Completing the New or Save As form](#)

- Rule resolution
- Requirements - Completing the Requirement Rules tab
- Requirements - Completing the Parameters tab

About Requirement Set rules

Purpose

RequirementSet Rules control how Requirement work objects are generated. They are referenced in the parent case so that at any time in the processing of the parent case, the Requirements specified in the Requirement Sets are generated. The parent case can be any work object (that is, any concrete class derived from Work-).

The Requirement Set custom rule form specifies:

- Class Name
- Requirement Rule Name

The Applies-When rule that evaluates whether a particular Requirement is generated

Access

Use the Application Explorer to access RequirementSet rules that apply to the work types in your application. Use the Records Explorer to list all RequirementSet rules available to you.

Category

Requirement Set rules are part of the RequirementSet category. They are instances of the RuleRequirementSet class.

Completing the New or Save As form

A Requirement Set rule has two key parts.

Field	Description
Label	Enter a name for this RequirementSet, starting with a letter and using only alphanumeric and dash characters. Choose a name that is unique within the Applies To class. The name must be a valid Java identifier. The length of the class name plus the length of the RequirementSet Name cannot exceed 128 bytes.
Applies To	<p>Identify the class to which this rule applies.</p> <p>Define the RequirementSet rules on the same or an ancestor class of the parent case class.</p> <p>The classes you enter here depend on the ruleset you select. On the Restrictions tab of the Class form, a class rule may limit rules applying to that class to belong to one of an explicit list of rulesets.</p>

Rule resolution

Full rule resolution applies to RequirementSet rules. When searching for RequirementSet rules, the system does the following:

Filters candidate rules based on a requestor's list of rulesets and versions Searches through ancestor classes in the class hierarchy for candidates when no matching rule is found in the starting class.

Requirements - Completing the Requirement Rules tab

This tab contains the fields described in this table.

Field	Description
Class Name	The Rule-Obj-Class rule on which the Requirement rule for this entry is defined.
Requirement Rule Name	The Requirement rule used at runtime to look at the Rule-Requirement definition and generate Requirement work objects.
Applies When	If not blank, names a When rule that is applied to determine if the requirement entry is applicable to a particular parent case at runtime based on properties in the parent case. The When rule is typically defined on the same class or an ancestor class of the parent case class.

Requirements - Completing the Parameters tab

This tab lists the parameters for the various Rule-Requirement rules on the Requirement Rules tab. The names are read-only, because the names of the parameters and their descriptions should be exactly the same as they are on the corresponding Rule- Requirement rules. You can edit the IN/OUT and Property columns. Below is a more detailed description of the columns on this tab.

This tab contains the fields described in this table.

Field	Description
Name	The name of the Parameter used to exchange data between the Requirement work object and the parent case.
Description	The description of the parameter used to exchange data between the Requirement work object and the parent case.
In/Out/Both	<p>Accepts parameters IN / OUT / BOTH.</p> <p>IN -Indicates that the parameter is passed IN to the parent case using this Requirement Set rule from the Requirement object at the time the requirement is validated.</p> <p>OUT- Indicates that the parameter is passed OUT to the requirement object from the parent case using this Requirement Set rule when the requirement is created.</p> <p>BOTH -Indicates that the parameter is passed OUT to the requirement object from the parent case using this Requirement Set rule when the requirement is created, and also passed IN to the parent case using this Requirement Set rule from the Requirement object at the time the requirement is validated.</p>
Property	The name of the property that contains the data to be mapped between the parent case and the parameter page. This property must be a single value scalar property that is

Field	Description
	present in the Applies To class of the Requirement rule definition or an ancestor class.

Pega Foundation for Financial Services requirements portal guide

- [About this document](#)
- [Overview](#)
- [Documents](#)
- [Requirements](#)
- [Requirement sets](#)
- [Simple example using the Requirements portal](#)

About this document

This document describes the core structures and configurations users will typically make within the Requirements functionality. Pega Foundation for Financial Services includes the Requirements Portal to make it easier to configure the Requirement Set and Requirement custom rules.

For guidance on the implementation of the underlying custom rule forms (Requirement Set and Requirement), see the *Pega Foundation for Financial Services Requirements Configuration Guide* on the product page.

- [Who should read this document?](#)

Who should read this document?

This guide is intended for those who want to understand the Requirements functionality, and how to use the Requirements Portal for design time configuration. The guide assumes a general knowledge of Pega® Platform, including an understanding of and the ability to create When, Activity, and Section business rules to customize the process and user experience.

Overview

The Requirements functionality provided in this product is highly configurable. The Requirements Portal is the preferred method for design time configuration because it provides a significantly improved user experience and eliminates the need to manually update any custom Rule forms.

Two underlying Custom Rule forms for Requirement and Requirement Set are in use. Those forms are documented in the more detailed *Pega Foundation for Financial Services Requirements Configuration Guide* on the [product](#) page.

- [Accessing the Requirements Portal](#)
- [Understanding and building the structures in Requirements](#)

Accessing the Requirements Portal

Users with PegaFSIF access group privileges have access to the Requirements Portal via the **Launch** menu in Designer Studio. The product provides the sample Business Analyst persona of *Akshay Davis* (user name akshay.davis@uplusfs.com). For more information on how to enable operators, see the *Pega Foundation for Financial Services Implementation Guide* on the [product](#) page.

Understanding and building the structures in Requirements

To enable reuse, the three core components of Requirement Sets, Requirements, and Documents exist as their own separate building blocks. A single type of Document can be used in different ways across multiple Requirements, and a Requirement can be used in different ways across multiple Requirements Sets.

At the appropriate stage in the process, the parent case will create a Requirement Set and the necessary subcases for each Requirement in that set. The Requirement Set structure allows the passing of parameters to and from the underlying Requirements work objects for use in processing. After each Requirement in the set is processed and satisfied, the parent process can continue.

Documents

Most organizations use many types of documents to support their business processes. For example, a customer may need to provide two forms of identification as part of a loan application to verify their identity.

To manage the types of documents that could be used within Pega processes, the Requirements Portal includes a Documents tab.

Name	Description	Last edited
435345345		
4506T	4506T	05/19/17
AddressProof		
Appraisal Report	Appraisal Report	02/07/18
Authorized Representative	Authorized Representative	
Bank Agreement	Bank Agreement	
BankingTermsAndConditions	Banking Terms and Conditions	
Birth Certificate	Birth Certificate	
Borrower EForm	Mortgage Application	09/24/14
Business Licence	Business Licence	

From this tab you can create, edit, and delete Document types to use in your processes. Groupings of Document Family and Document Category are provided to enable managing large libraries of document. Each Document has a Referencing Requirements section that you can expand to show which Requirements in the system are using the Document. You cannot delete a Document that is referenced by other Requirements.

Update Document Details X

Document: State ID

Document Family*	Application layer Pega Foundation for Financial Services 8.1
Loan Application	
Document Category*	History details
General Documentation	Added by Updated Oct 11, 2017 by FSIFSYSADMIN
Document name	Referencing requirements
StateID	Requirements found: 3
Description	Rule name
Identification - State	<u>MortgagePersonalIdentification</u>
Type*	MortgagePersonalIdentification
Manual	ProofOfResidence

Cancel Update

Note: The previous figure demonstrates the creation of a Document type.

- ⓘ Actual documents of this type will be created at runtime and stored in either Pega 8 or a Content Management System such as Alfresco® or IBM® FileNet®.

There are three main formats of Documents available, which once created, can be used alone or in combination with other Documents in a Requirement.

- **Manual** - The simplest document format of the three, it is used to represent an attachment of an electronic copy of a document such as a scanned image of a credit report or driver's license.
- **Binary** - Allows Pega to generate an existing instance of a Binary File Rule when the document is required in the process. No data can be passed to the document at generation time. These are typically items such as a Terms and Conditions document or a standard agreement that a customer would sign and return. The core parameters of Location, File Name, and Extension from the underlying Binary Rule Form are required when a Binary Document is added to the library.

Type*

 Binary file

 Location File Name Extension
 _____ _____ _____

- **eForm** - An eForm or Electronic Form allows data to be populated in the document when generated at runtime. For example, a Loan Application Form could be populated with specific applicant details that the organization has on file. This speeds up the application process and prevents applicants from having to manually fill out a form. An existing eForm Map Rule must be specified when an eForm Document is added to the library.

Type*

 eForm map rule

 MortgageURLAeForm

Requirements

A Requirement refers to one or more actions that must be completed before proceeding to the next step in a process. Examples of Requirements might include: verification of identity for a loan application, completion of an application form, or proof that a survey call has taken place. A Requirement typically pulls together one or more documents to be collected and verified in a variety of ways, but it does not have to include a document.

The following figure shows an example of a Personal Identification Requirement:

- **Requirement types**
- **Requirement configuration settings**

Requirement types

There are two types of Requirements available: Guided Documentation Validation, and Document Group.

- **Guided document validation**
- **Document group**
- **Adding documents**

Guided document validation

Guided Document Validation provides a full set of functionalities to guide users through satisfaction of the Requirement. This includes but is not limited to:

- Providing verification instructions to capture data against a document or flow (e.g., prompt a user to enter data from the document they are viewing).
- Passing data to and from the parent Work Object during processing of the Requirement (e.g., previous Credit Score).
- Running Validation rules to check data captured against each document and then cross-referencing the overall Requirement level between them (e.g., Date of Birth).

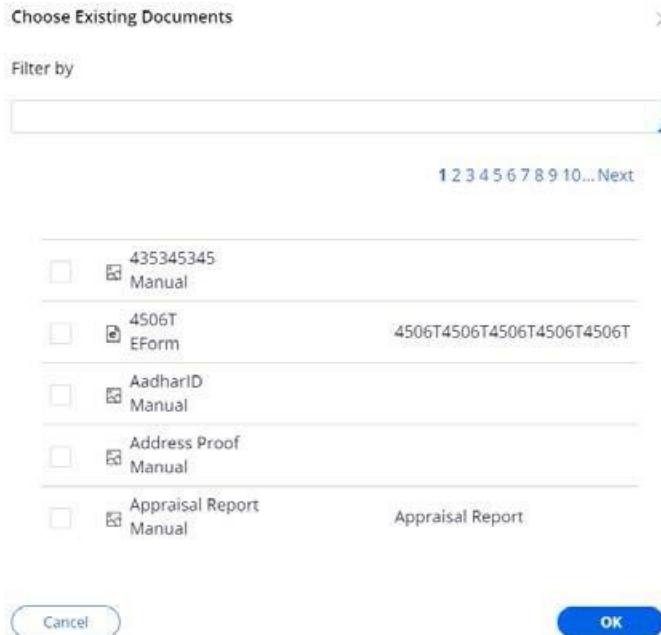
Document group

A Document Group is a list of documents to collect at runtime without any verification guidance or validation checks. The user is prompted to upload the equivalent documents and manually mark them as In Good Order. When a Document Group is used at runtime, no Requirements subcases are created.

This requirement type is used during Customer onboarding in the Onboarding for Financial Services application.

Adding documents

To add Documents to a Requirement, click **Add** under the Documents section of the Requirement screen. The Choose Existing Documents dialog box opens:



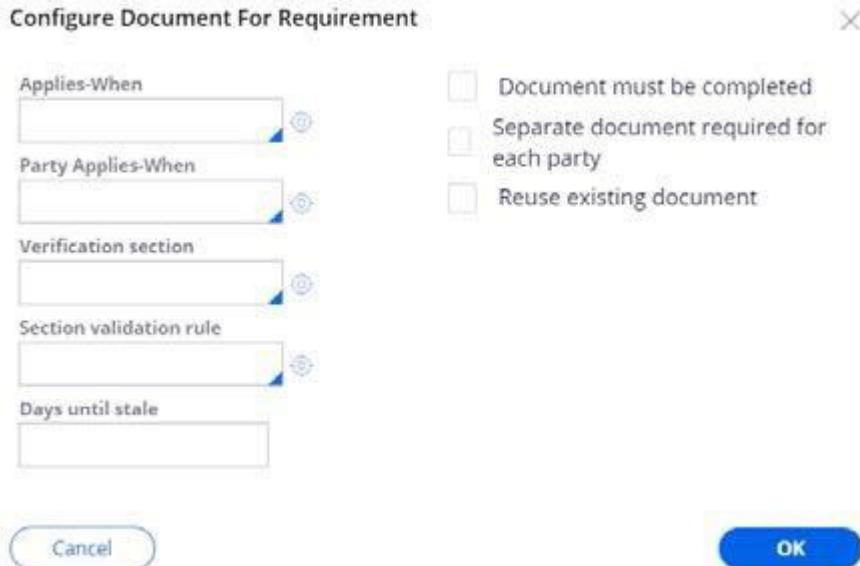
Users can view each page of documents and select one or more documents from the list. Use the **Filter by** field to filter the search results.

Once Documents have been added to the Requirement, they appear on the main Requirement screen. To configure how these documents are handled at runtime, click the Configuration icon (⚙️) for the desired Document. A warning triangle (⚠️) indicates that users should review the Document configuration settings. The warning no longer appears once the configuration settings have been accessed.

The screenshot shows the Requirements screen with a "Documents" header and an "Add" button. Below the header is a table-like structure with two rows:

State ID Manual	Identification - State	
Birth Certificate Manual	Birth Certificate	

The following figure shows the available configuration options for a Document within a Requirement. The fields include tooltips that explain each field.



- **Applies-When** - Identifies a When rule that determines whether this Document type is applicable to a particular case. For example, a form might be applicable in some states but not others.
- **Party Applies-When** - Identifies a When rule that determines whether this Document entry is applicable for a type of party on the case. For example, a Document may be applicable for a primary applicant but not for a secondary applicant.
- **Verification Section** - Identifies a Section rule to show when the user clicks the **Validate** link for this Document at runtime. The Section guides them to capture or confirm information that can then be used in processing the Document. For example, the user is prompted to enter the **Date of Birth** and **Name** on the passport provided by the customer.
- **Section Validation Rule** - Identifies the accompanying Validate rule used to validate information captured in the Verification Section for the Document. For example, has the monthly income specified increased over a specified threshold since it was last captured?
- **Days until Stale** - The number of days after which a Document will be considered *stale* meaning that it will no longer be marked as **In Good Order**, and it needs to be processed again.

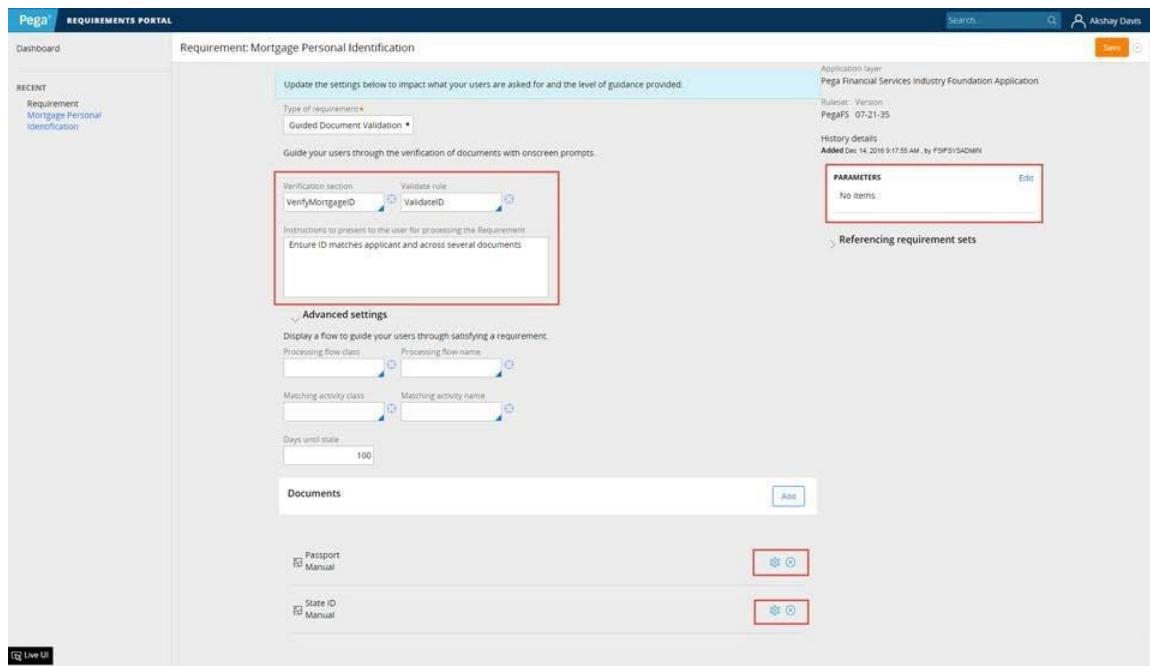
- **Document must be completed** - When selected, the Requirements cannot be satisfied unless this document has been collected and has a status of **In Good Order**. Do not select this check box for optional Documents in a process.
- **Separate Document required for each party** - Select this check box if the parent case has multiple parties.
- **Reuse existing document** - If selected, at runtime the application searches the system for an existing document that satisfies this entry and, if found, links it to this requirement.

Note: The Reuse existing document check box is for legacy FSIF versions only.

The current CMIS integration automatically searches for existing documents.

Requirement configuration settings

The Requirement screen in the Requirements Portal offers a wide range of configuration options. The following figure shows an example of the Requirement and its configuration settings.



There are three areas of settings on the Requirement screen:

- **Requirement specific settings** - These settings control overall processing of the Requirement. The advanced settings are hidden by default.
 - **Document specific settings** - These are described in the previous section of this document. These settings control how a document is processed in the context of the Requirement that includes it. Base types of Documents such as Passport or State ID can be used in different ways across multiple Requirements. After you add a document to the Requirement, these settings are accessed separately for each Document via the Configuration icon (⚙).
 - **Parameters** - Allow data to be passed to and from the parent Work Object via the Requirement Set that includes them. These settings are accessed via the Parameters Panel on the right-hand side of the Requirement screen.
- **Requirement-specific settings**
- **Parameters**

Requirement-specific settings

- **Verification Section** - If this field is populated, this optional Section rule is displayed only after individual documents in the Requirement have been validated. The purpose is to provide an additional level of validation for the whole Requirement. An example might be asking a user to confirm further information or showing the user a screen that compares information captured when verifying two separate Documents.
- **Validation Rule** - This is the accompanying Validate rule used to validate information captured in the Verification Section for the overall Requirement.
- **Instructions to present to the user for processing the Requirement** - These instructions will appear to the user when the Requirement is presented at runtime.
- The following additional settings are available from the **Advanced Settings** link:
 - **Processing Flow Name** - Identifies an alternate flow to use to process the Requirement, instead of using the standard flow provided.

- **ProcessingFlow Class** - The class of the alternative flow.
- **Matching Activity Name** - In cases where a Requirement is manually added to a Requirement Set at runtime, identifies the activity that verifies that the Requirement Set is not already in place so as to avoid duplicates.
- **Matching Activity Class** - The class of the alternative matching activity specified.
- **Days Until Stale** - The number of days after which the Requirement will be considered *stale* meaning that it will no longer be *satisfied*, and it needs to be processed again.

Parameters

The screen for editing or adding a new Parameter can be accessed via the Edit link in the Parameter panel on the right side of the Requirement screen.

The screenshot shows the 'Application layer' interface for a 'Ruleset' named 'PegaFSRequirements' and 'Version' '08-01-01'. It includes 'History details' showing an item was 'Added' on 'Jul 4, 2018 8:27:01 AM' by 'Rakesh Kumar'. A 'Parameters' panel is open, displaying 'No items' and an 'Edit' link. Below the panel, a link 'Referencing requirement sets' is visible.

When a Requirement with parameters is included in a Requirement Set, equivalent configuration settings are shown on the Requirement Set screen.

Add/Edit Parameters

Input activity :	<input type="text"/>	Output activity :	<input type="text"/>	
Name	In/Out	Property	Page?	Page class
<input type="text"/> Name	in	<input type="text"/> .FullName	<input type="checkbox"/>	<input type="button" value="X"/>
<input type="text"/> Address	in	<input type="text"/> .pyCity	<input type="checkbox"/>	<input type="button" value="X"/>

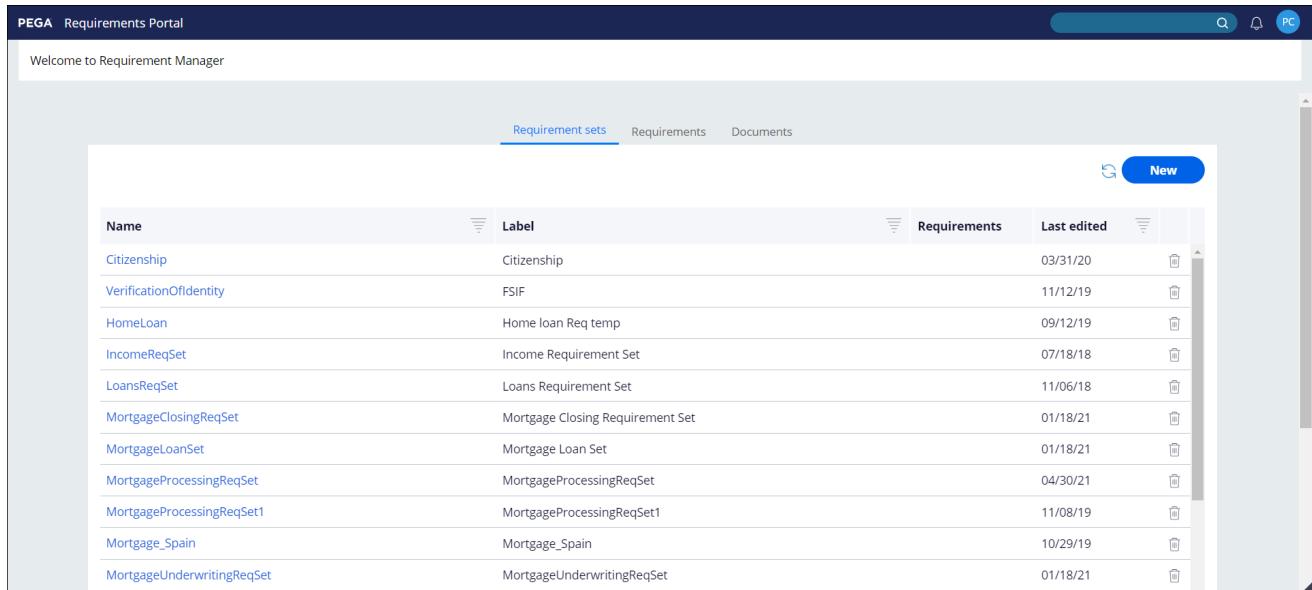
The figure above shows an example of a Requirement with parameters. To add new rows to the list:

1. Click the Add icon (+).
2. Enter the **Name**.
3. Select whether the parameter is in, out, or both.
4. Enter the **Property** that the parameter will be mapped to.
5. Click **OK**.

Optional Activities to be executed for input and output can be specified in the fields provided.

Requirement sets

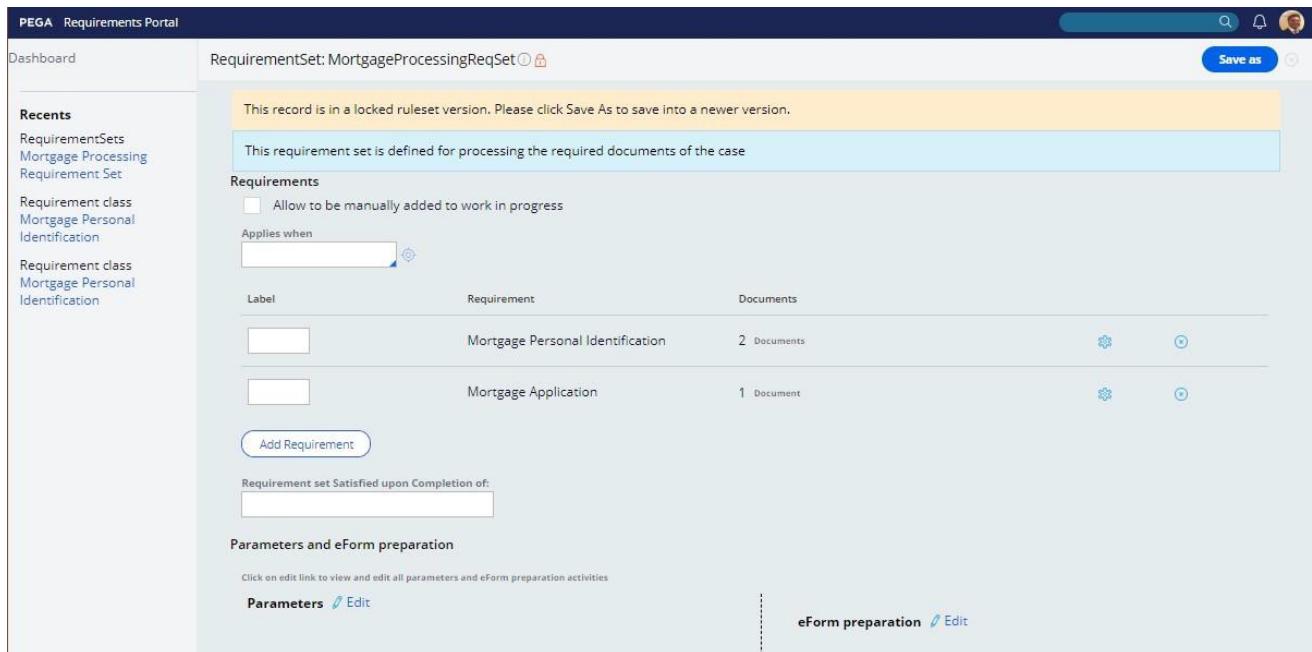
Requirement Sets specify Requirements and the settings that determine when they apply within the context of the Pega case they are being used within. Requirement Sets can be managed from the Requirement Sets tab on the Requirements Portal.



The screenshot shows the PEGA Requirements Portal interface. At the top, there's a navigation bar with tabs for 'Requirement sets', 'Requirements', and 'Documents'. A 'New' button is located in the top right corner. Below the navigation is a table listing various requirement sets. The columns are labeled 'Name', 'Label', 'Requirements', and 'Last edited'. Each row contains a small trash icon in the last column.

Name	Label	Requirements	Last edited
Citizenship	Citizenship		03/31/20
VerificationOfIdentity	FSIF		11/12/19
HomeLoan	Home loan Req temp		09/12/19
IncomeReqSet	Income Requirement Set		07/18/18
LoansReqSet	Loans Requirement Set		11/06/18
MortgageClosingReqSet	Mortgage Closing Requirement Set		01/18/21
MortgageLoanSet	Mortgage Loan Set		01/18/21
MortgageProcessingReqSet	MortgageProcessingReqSet		04/30/21
MortgageProcessingReqSet1	MortgageProcessingReqSet1		11/08/19
Mortgage_Spain	Mortgage_Spain		10/29/19
MortgageUnderwritingReqSet	MortgageUnderwritingReqSet		01/18/21

The following figure shows an example Requirement Set for the Processing Stage of a Loan.



This screenshot shows the configuration of a requirement set named 'MortgageProcessingReqSet'. The left sidebar lists recent requirement sets and requirement classes. The main area displays the requirement set details, including its purpose for processing required documents and its requirements. One requirement is listed: 'Allow to be manually added to work in progress'. The 'Parameters and eForm preparation' section indicates that parameters and eForm preparation activities can be edited.

- Adding requirements
- Setting parameters
- Setting eForm preparation

- Applicability and satisfaction logic

Adding requirements

To add Requirements to a Requirement Set, click **Add Requirements** under the Requirements section of the Requirement Set screen. The following dialog box opens:

The dialog box has a header 'Choose Existing Requirements' with a close button. Below it is a 'Filter by' input field. A navigation bar at the bottom shows page numbers from 1 to 9 followed by 'Next'. The main area contains a table with columns: Select, Name, Description, and Class name. The table lists several requirements:

Select	Name	Description	Class name
<input type="checkbox"/>	Bank Agreement	Bank Agreement	PegaReq-Data-ReqVerify-
<input type="checkbox"/>	BusinessAddress	Business Address	PegaReq-Data-ReqVerify-
<input type="checkbox"/>	BusinessLicense	Business License	PegaReq-Data-ReqVerify-
<input type="checkbox"/>	Citizenship	Citizenship	PegaReq-Data-ReqVerify-
<input type="checkbox"/>	Company Constitution Document	Company Constitution Document	PegaReq-Data-ReqVerify-

At the bottom are 'Cancel' and 'Submit' buttons.

Users can view each page of Requirements, selecting one or more Requirements from the list. Use the **Filter by** field to filter the search results.

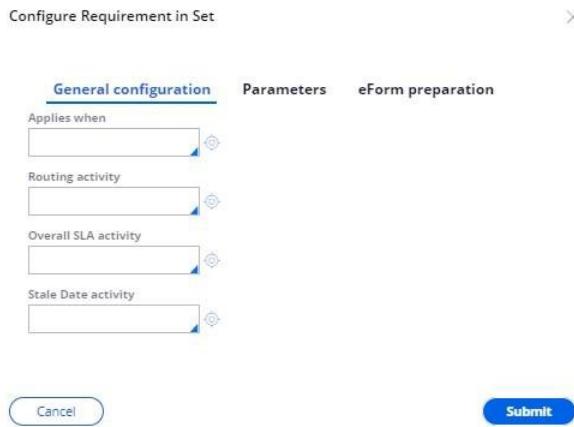
Once Requirements have been added to the Requirement Set, they appear on the main Requirement Set screen. To configure how these Requirements are handled at runtime, click the Configuration icon (⚙) for the desired Requirement. A warning triangle (⚠) indicates that users should review the Requirement configuration settings. The warning no longer appears once the configuration settings have been accessed.

The table has columns: Label, Requirement, and Documents. It lists two requirements:

Label	Requirement	Documents
<input type="checkbox"/>	Mortgage Application	1 Document
<input type="checkbox"/>	Mortgage Personal Identification	2 Documents

An 'Add Requirement' button is located at the bottom left.

The following figure shows the available configuration options for a Requirement within a Requirement Set. The fields include tooltips that explain each field.



- **Applies-When** - A When rule used to determine whether a Requirement is generated.
- **Routing activity** - An activity to route the Requirement to a workbasket or worklist at runtime based on specific case data.
- **Overall SLA Activity** - Defines an overall SLA on this Requirement.
- **Stale Date Activity** - Once a requirement becomes stale, certain processing can be allocated using this activity.

Setting parameters

As noted earlier in this guide, Requirements can be configured to contain parameters. These parameters pass data from and to the parent Work Object for processing. When a Requirement containing one or more Parameters is included in a Requirement Set, you can set the equivalent configuration on the Requirement Set in two ways:

- Click the Configuration icon (⚙️) for the Requirement, and then select the Parameters tab in the dialog box that opens. The tab shows only Parameters that are specific to the chosen Requirement.
- Click the **Edit** link next to **Parameters** at the bottom of the Requirement Set screen. The screen shows all parameters for all Requirements included in the Requirement Set.



You can use the default parameter values from the corresponding Requirement, or update the parameter values to reflect differing mapping needs.

- Note:** The Parameter Name is a read-only value because the name is common to every Requirement Set that the Requirement containing the parameter was created in.

State ID					
Name	Description	In/Out	Property	Page?	Page class
ID	ID	in	IDs		

Cancel Update

Setting eForm preparation

As described earlier, a Document can be of the type *eForm*. When a Requirement containing a Document of type eForm is included in a Requirement Set, you can configure an activity at the Requirement Set level to carry out pre-processing before the eForm data is used. You can do this in two ways:

- Click the Configuration icon (⚙) for the Requirement, and then select the eForm Preparation tab in the dialog box that opens. This displays only eForms specific to the chosen Requirement.
- Click the **Edit** link next to **eForm Preparation** at the bottom of the Requirement Set screen. The screen shows all eForms for all Requirements in the Requirement Set with a Document of type eForm.

Parameters and eForm preparation

Click on edit link to view and edit all parameters and eForm preparation activities

Parameters [Edit](#)

eForm preparation [Edit](#)

- If a preparation activity is required, you can enter it from either link.

eForm Preparation

Mortgage Application

Name	Description	Preparation activity
Mortgage Application	Uniform Application form 1003	<input type="text"/>

State ID

Name	Description	Preparation activity
Signature Card 01	Signature Card 01	<input type="text"/>
W9	W9	<input type="text"/>

[Cancel](#) [Update](#)

Applicability and satisfaction logic

A When rule can be specified in the **Applies when** field to allow a process to determine when a Requirement Set should be used or not.

Requirements

Allow to be manually added to work in progress

Applies when

Label	Requirement	Documents
A	Mortgage Application	1 Document
B	Mortgage Personal Identification	2 Documents

Satisfaction logic can be set for the Requirements in the Requirement Set. This uses the same format as criteria in a Report Definition. The user specifies a letter for each Requirement and can then add logic to the **Requirement set satisfied upon completion of** field. For the example below the whole Requirement Set could be satisfied if only one of the Requirements in the set was satisfied.

Label	Requirement	Documents		
A	Mortgage Application	1 Document		
B	Mortgage Personal Identification	2 Documents		
C	State ID	3 Documents		

Add Requirement

Requirement set Satisfied upon Completion of:
(A OR B) AND C

Simple example using the Requirements portal

The Requirements functionality comes with the Apply for Personal Loan sample. The sample flow includes three Requirement Sets, seven Requirements, and seven Documents. You can explore these assets in the Requirements Portal to see how they relate to the runtime experience.

To help familiarize you with the core design-time approach, this topic walks through a simple example in the Requirements Portal.

To get started, log in to the Designer Studio as a user with access to the Requirements Portal. Select the Requirements Portal from the **Launch** menu.

- **Create document**
- **Create requirement**
- **Create a Requirement set**
- **Review your created assets**

Create document

1. Click **Create New** from the Documents section on the right side of the dashboard.

The screenshot shows the PEGA Requirements Portal interface. At the top, there's a dark header bar with the PEGA logo and "Requirements Portal". To the right of the header are search, notification, and PC selection icons. Below the header, a banner says "Welcome to Requirement Manager". Underneath the banner, there are three tabs: "Requirement sets", "Requirements", and "Documents", with "Documents" being the active tab. A large blue "New" button with a circular arrow icon is positioned at the top right of the main content area. The main content area displays a table with four columns: "Name", "Description", and "Last edited". There are four rows in the table:

Name	Description	Last edited
435345345		
4506T	4506T	05/19/17
AddressProof		
Appraisal Report	Appraisal Report	02/07/18

2. Enter the document information as shown below, and then click Create.

Add New Document X

Document Family ★
Loan Application

Document Category ★
General Documentation

Name ★
Birth Certificate

Description
Birth Certificate

Type ★
Manual

Cancel Create

After you successfully add a new document, the main dashboard window shows a confirmation message. Enter the name of new Document in the search field and confirm the new Document name appears in the list.

The screenshot shows the PEGA Requirements Portal dashboard. The top navigation bar includes a search icon, a bell icon, and a PC icon. Below the bar, the text "Welcome to Requirement Manager" is displayed. The main area features three tabs: "Requirement sets", "Requirements", and "Documents". The "Documents" tab is selected, and a blue "New" button is located in the top right corner of the list area. The list displays columns for "Name", "Description", and "Last edited". One entry, "Birth Certificate", is listed with "Description" as "Birth Certificate" and "Last edited" as the current date.

Create requirement

1. Click **Create New** in the Requirements section of the Requirements Dashboard.

The screenshot shows the PEGA Requirements Portal dashboard. The top navigation bar includes a search icon, a bell icon, and a PC icon. Below the bar, the text "Welcome to Requirement Manager" is displayed. The main area features three tabs: "Requirement sets", "Requirements", and "Documents". The "Requirements" tab is selected, and a blue "New" button is located in the top right corner of the list area. The list displays columns for "Name", "Label", "Documents", and "Last edited". Two entries are listed: "Mortgage Application" (Label: Mortgage Application) and "MortgageClosingAgentContact" (Label: Closing Agent Contact).

2. The Add New Requirement screen opens. Create a new *BirthPlace* requirement as shown in the following figure.

The screenshot shows the "Add New Requirement" dialog box. At the top, there are "Create" and "Cancel" buttons. The form fields include:

- Name ***: BirthPlace
- Label ***: Place of Birth
- Application/RuleSet** (under Application layer):
 - Pega Foundation for Financial Services 8.1
 - Pega Accounting 8.1
 - Pega Requirements 8.1
 - UI Kit
 - Pega Platform
- PegaReq-Work-ProcessRequirement**
- View all**
- Ruleset ***: A dropdown menu with an error message: "Value cannot be blank".
- Version ***: A dropdown menu.

 A note at the bottom states: "No unlocked rulesets available".

3. For the Name, enter ***BirthPlace***. For the Description, enter ***Place of Birth***. Select which Application Layer, ApplyTo Class, Ruleset, and Ruleset Version you want the underlying Requirement rule to be created in, and then click Create.



Note: You must have an open ruleset version in which to create new Requirements.

4. On the next screen, select Guided Document Validation as the Type of Requirement.

- Add a document to the requirement
- Configure the document within the requirement
- Add a parameter

Add a document to the requirement

Click Add in the Documents section to open the Choose Existing Documents dialog box. Select the BirthCertificate document from page 2 in the list, and then click OK.

Choose Existing Documents

Filter by

Previous 1 **2** 3 4 5 6 7 8 9 10... Next

<input type="checkbox"/> AuthorizedReps Manual	AuthorizedReps
<input type="checkbox"/> AuthSignersDoc Manual	AuthSignersDoc
<input type="checkbox"/> BankAgreement Manual	BankAgreement
<input checked="" type="checkbox"/> BirthCertificate Manual	Birth Certificate
<input type="checkbox"/> BORROWERCREDITREPORT Manual	BORROWERCREDITREPORT

Cancel **OK**

The PlaceOfBirth Requirement now includes the BirthCertificate Document. The warning triangle icon () indicates that configuration can take place.

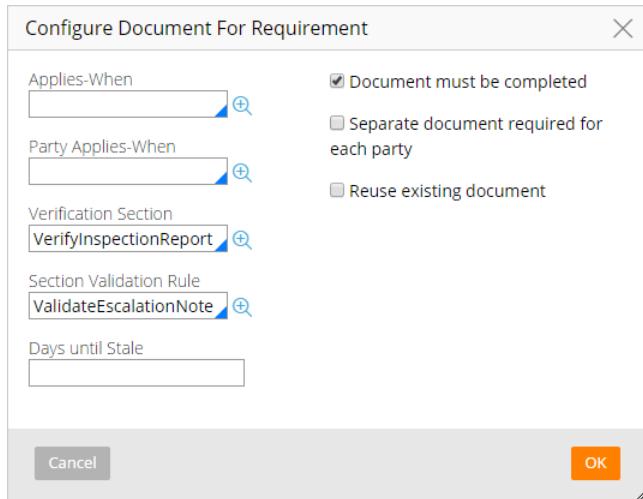
Documents

Add

BirthCertificate Manual	Birth Certificate	
-------------------------	-------------------	--

Configure the document within the requirement

Click the Configuration icon () to show the configuration settings. Add some sample configuration sections as shown below, and then click **OK**.



- Note:** If this were a real implementation, you would select the necessary When, Section, and Validate rules to customize the process and user experience.

Add a parameter

1. Click the Edit link in the Parameter panel of the Requirement screen.

Parameters	Edit
No items	

2. Click the Add icon (⊕) to create two new parameters—BirthNameFull and BirthCity—as shown in the following figure. Enter the appropriate properties to map to the parameters.

Add/Edit Parameters

Input Activity :		Output Activity :	
Name	In/Out	Property	Page?
BirthFullName	in	.pyWorkParty(Customer)	<input type="checkbox"/>
BirthCity	in	.pyCity	<input checked="" type="checkbox"/>
<input style="width: 100px; height: 20px; margin-bottom: 10px;" type="button" value="+"/> <input style="margin-right: 10px;" type="button" value="Cancel"/> <input style="width: 50px; background-color: orange; color: white; border-radius: 5px;" type="button" value="OK"/>			

- Click OK. The new Parameters appear in the left panel as shown below.

Parameters

<input style="width: 50px; height: 20px; margin-bottom: 10px;" type="button" value="Edit"/>
BirthFullName
BirthCity

- Click Save to save the BirthPlace Requirement. You receive a success message, and the new Requirement becomes available in the list on the Requirement Dashboard under the Description Place of Birth.

Create a Requirement set

- Click New in the Requirement Sets tab of the Requirements Dashboard.

PEGA Requirements Portal

Welcome to Requirement Manager

Requirement sets				
Name	Label	Requirements	Last edited	
Citizenship	Citizenship		03/31/20	
VerificationOfIdentity	FSIF		11/12/19	
HomeLoan	Home loan Req temp		09/12/19	
IncomeReqSet	Income Requirement Set		07/18/18	
LoansReqSet	Loans Requirement Set		11/06/18	

- The Add New Requirement Set screen opens. For the Name, enter *Loan Onboarding*. For the Label, enter *Loan Onboarding*. Select which Application Layer,

ApplyTo class, RuleSet, and RuleSet Version you want the underlying Requirement set rule to be created in, and then click Create.



Note: You must have an open ruleset version in which to create new Requirements.

The resulting screen shows your new Requirement Set.

The screenshot shows the PEGA Requirements Portal interface. At the top, it says "PEGA Requirements Portal" and "RequirementSet:LoadOnboarding". Below that, a note states: "This requirement set is defined for processing the required documents of the case". Under the "Requirements" section, there is a checkbox "Allow to be manually added to work in progress" which is checked. A "Label" field is present, and below it, a "Requirement" table with one row labeled "No items". An "Add Requirement" button is visible. Further down, there is a "Requirement is satisfied by:" field containing "e.g. A OR B if labels are A," and an "Optional description of satisfaction:" text area with a rich text editor toolbar. A note at the bottom of this area says: "If provided, description is shown to users instead of automated description."

3. Click Add Requirement on the Requirements tab. Enter *Identity* in the Filter by field, select Proof of Residence from the list of matches, and then select the Proof of Residence Requirement.

Choose Existing Requirements

Filter by
IdentityProof

Select	Name	Description	Class name
<input checked="" type="checkbox"/>	IdentityProof	Proof Of Residence	PegaReq-Data-ReqVerify-

Cancel **Submit**

- Click Submit, and then notice that the Identity Proof Requirement is added.

PEGA Requirements Portal

RequirementSet:LoadOnboarding

This requirement set is defined for processing the required documents of the case

Requirements

Allow to be manually added to work in progress

Applies when

Add Requirement

Label	Requirement	Documents
	Proof Of Residence	1 Document

Requirement is satisfied by:
e.g. A OR B if labels are A,

Optional description of satisfaction:

If provided, description is shown to users instead of automated description.

- Click the Configuration icon (•) to the right of the Place of Birth Requirement to display the configuration options. Select the Parameters Tab. The parameters from the Place of Birth Requirement are available. In this example there are none.

Configure Requirement in Set

Name	Description	In/Out	Property	Page?	Page class
No items					

Cancel **Submit**

- Click Cancel to close the Configure Requirement Set dialog box. Click Save to save your Requirement Set, and then click X to close the Requirement Set and return to the Requirements Dashboard.

Review your created assets

If the LoanOnboarding Requirement Set is not in place, click the refresh icon (↻) on the Requirement Set tab. Open the new Requirement Set to review that the Loan Onboarding Requirement Set contains a Requirement called Proof of Residence, which in turn contains one Document called *Proof Of Residence*.

These assets can now be used as part of the implementation of your application. For more information, see the *Pega Foundation for Financial Services Requirements Configuration Guide* on the product page.

Pega Foundation for Financial Services Version Verification Tech Note

- **Introduction**
- **Verification**

Introduction

This tech note is helpful for verifying the Pega Foundation for Financial Services version number.

Verification

Verify version of Pega Foundation for Financial Services (PFFS) former Pega Financial Services Industry Foundation (FSIF) by looking at the version listed on the label in the application rule

Note:

- Every application rule for 7.4 release and higher will have version specified in the application rule label
- Application(s) listed under Built on applications will only have major version (Example: 8) specified.

For Example: Application rule for **Sample Application for PFFS 8.4**. Version is specified in the label.

Name	Version
PegaFSIF	8

Example: Application rule for **Foundation for Financial Services 8.4**. Version is specified in the label.

Edit Application: Foundation for Financial Services 8.4
ID: PegaFSIF • 8 RS: PegaFS [Edit]

This record has 3 info warnings (including 3 unjustified) [Review/Edit](#)

[Delete](#) [Actions](#) [Save](#) [X](#)

Definition Cases & data Application wizard Documentation Integration & security History

Password Supply password to update	Development branches + Add branch No items										
Built on applications + Add application <table border="1"><thead><tr><th>Name</th><th>Version</th></tr></thead><tbody><tr><td>1 PegaAccounting</td><td>8</td></tr><tr><td>2 PegaRequirements</td><td>8</td></tr></tbody></table>	Name	Version	1 PegaAccounting	8	2 PegaRequirements	8	Application rulesets + Add ruleset <table border="1"><tbody><tr><td>1 PegaFSIF-HealthCheck:08-04</td></tr><tr><td>2 PegaFSWeb:08-04</td></tr><tr><td>3 PegaFSRequirements:08-04</td></tr><tr><td>4 PegaFS:08-04</td></tr></tbody></table>	1 PegaFSIF-HealthCheck:08-04	2 PegaFSWeb:08-04	3 PegaFSRequirements:08-04	4 PegaFS:08-04
Name	Version										
1 PegaAccounting	8										
2 PegaRequirements	8										
1 PegaFSIF-HealthCheck:08-04											
2 PegaFSWeb:08-04											
3 PegaFSRequirements:08-04											
4 PegaFS:08-04											
Enabled components + Add component											

In older versions of PFFS (former FSIF) version is not specified in the application rule. In this case, click on gadget icon next to PegaFS: xx-xx ruleset in the application stack.

Verify version against a chart of PegaFS rulesets:

Pega Foundation for Financial Servicesversion	Rulesetversion
7.21	PegaFS:07-21-01
7.21.1	PegaFS:07-21-11
7.22	PegaFS:07-21-35
7.31	PegaFS:07-21-40
7.32	PegaFS:07-21-45
7.4	PegaFS:07-21-50
8.1	PegaFS:08-01-01
8.2	PegaFS:08-02-01
8.3	PegaFS:08-03-01
8.4	PegaFS:08-04-01
8.5	PegaFS:08-05-01

Pega Foundation for Financial Servicesversion	Rulesetversion
8.6	PegaFS:08-06-01
8.7	PegaFS:08-07-01

Resources

The following table lists the PDFs/Excel files of Pega Foundation for Financial Services for '23

Document type	Format
Pega Foundation for Financial Services data dictionairy	XLSX