Enterprise Correspondence Management - Letters

For development specific information, view ECM-L Developer View.

For UI/UX design specific information, view ECM-L UI/UX Design View.

For General Audience specific information, view ECM-L General Audience View.

Solution Metadata

Name	*Value*
Solution Status	*DRAFT*
Solution Type	*NEW TENANT*
Tenant	*TBD ECM-L*
Recommended Namespace	*ecml*
Recommended BIP Tenant App(s)	*bip-ecml-resource-manager* *bip-ecml-resource-manager-mfe-ui* *bip-ecml-letter-manager* *bip-ecml-letter-manager-mfe-ui* *bip-ecml-template-manager* *bip-ecml-template-manager-mfe-ui* *bip-ecml-svc* *bip-ecml-mfe-root-ui*
Sponsoring organization	*OIT*
Number of users	*TBD*
Estimated Monthly Cost	*TBD*
Privacy	*Differs from system to system.* *Resource Manager - None.* *Template Manager - None.* *Letter Manager - Both possible.* *ECM-L-API - Both possible.*
Will need 508 Compliance	*Yes, all of the MFE UI's will require 508 compliance.*
Deployment date	*TBD*

Introduction

This document outlines the design for Enterprise Correspondence Management-Letters (ECM-L), a centralized platform that empowers business users to efficiently manage letter content and generation. Currently, multiple Lines of Business (LoBs) within the Veterans Affairs infrastructure use disparate methods and applications to create correspondence, resulting in operational inefficiencies and a heavy dependence on IT support.

ECM-L addresses these challenges by providing a unified solution that enables LoBs to self-manage their correspondence needs within a standardized environment. Once onboarded, LoBs can independently make changes—such as updating Letter Templates and Resources—without requiring IT intervention. This centralized approach provides a controlled, reliable environment for producing, testing, releasing, and storing correspondence while streamlining operations and enhancing consistency across all LoBs.

What Problem Needs to be Solved?

Individual Lines of Business (LoBs) within the Veterans Affairs Organization utilize disparate methods and applications to generate their own correspondence. Without a centralized Enterprise Correspondence Management system, each LoB operates independently for Letter Templates and changes, and any substantial changes to Templates or mandates rely on IT resources for those changes, often requiring customized development for each LoB. This fragmented approach creates resource bottlenecks, operational inefficiencies, and limited historical tracking, ultimately resulting in delays and potential inaccuracies in the correspondence provided to our Veterans.

Business Needs

Is this a defect or an enhancement?	Enhancement
Who reported the need?	Office of Information & Technology (OIT)
When was the need noticed?	Q2 2022
When did/will it become a problem?	Q2 2024
Is there an existing Requirement? Strategic Objective?	Enterprise Correspondence Management (ECM) Expansion | BPT-4700
What existing resources, tickets, emails, etc. might serve to better understand the problem?	Namespace and Tenant App Request: • ECM-L | BPS-74504 • Resource Manager API | BPS-69060 • Template Manager API | BPS-69063 • Letter Manager API | BPS-69064 • ECM-L API | BPS-69085 Tenant Security Assessment (TSA) Request under BIP ATO: • ECM-L | VBMSD-337002
What business processes are involved?	Document Generation, Document Storage and Retrieval
Who is the business owner for the impacted systems or processes?	• Jeffry Boutet • Brian Steege • Peter Graziatis
Who are the business decision makers for solving the problem?	Pam Devine
What is the historical context? How did we get where we are?	Historically, VA has used multiple applications to generate, manage, and edit letters. Correspondence Service has provided document generation functionality to the VBMS system, providing document generation for use in VBMS Core, Ratings, and Awards. With this architecture, any changes needed by the business specific letter requires large amounts of commitment and OIT resources.
What happens to the business stakeholders if nothing is done?	VA will have disjointed efforts and applications for correspondence and letters
Of the various aspects of this problem, which of them does the business consider as critical flaws? Which are nice-to-have changes?	• Need for improvements in Letters Finalization • Need for centralized enterprise system that allows multiple LoBs to draft, generate, and Finalize Letters

Costing Considerations

Description	Summary of Costs	Total
Initial Development	• Staffing costs: **$7.2M** • Tools and licenses: **$0.7M** • Training: **$8.4M** • Contingency: **$0.2M**	**$8.4M**

AWS GovCloud Infrastructure	• Compute: **$0.4M / Year** • Storage: **$0.3M / Year** • Database: **$0.1M / Year** • Networking/Other: **$0.1M / Year**	**$0.9M / Year**
Annual Maintenance	• Staffing costs: **$1.2M / Year** • Infrastructure updates: **$0.3M / Year** • License renewals: **$0.1M / Year** • Ongoing enhancements: **$0.5M / Year**	**$2.1M / Year**
5-Year Total Cost of Ownership	• Development: **$8.4M** • Infrastructure: **$4.5M** ($0.9M Per Year x 5 years) • Maintenance: **$9.5M** ($2.1M Per Year x 4.5 years)	**$22.4M**

Chargeback Considerations

ECM-L creates natural cost separation by organizing all of its components into distinct buckets for each Line of Business (LOB). We will be able to evaluate and isolate usage metrics based on the individual Lines of Business to inform and streamline chargeback considerations.

Technical Needs

Historically, VA has used multiple applications to generate, manage, and edit letters. This causes numerous workflow inconsistencies, various vendors, and large amounts of OIT commitments and resources, none which seem to have a perfected process in place. The purpose of Enterprise Correspondence Management- Letters (ECM-L) is to create a centralized, enterprise system, built around shared concepts, processes, and workflows to be leveraged by multiple Lines of Business (LoBs) in the Veterans Benefits Administration (VBA). The system will be LoB oriented, each being responsible for their integration and usage, with BID only supporting, facilitating, and trouble-shooting as needed. Onboarding new LoB will not require development work, but use configurations and system activities.

In order to prevent unintended interactions or dependencies, as well as support charge-back capabilities, the content, processes, and data of each LoB will be completely separate from those of another LoB. In addition, the ECM-L will be LoB agnostic, providing reasonable common denominators in concepts or processes among the LoBs, but ECM-L will not contain any proprietary business logic or concepts. Each LoB will be able to integrate and incorporate their own business logic or processes through predefined hooks, APIs, or other integration points. This will allow API access to Letter Instances in data format as well as give LoBs the ability to configure data lookup "callback" to its own systems.

The implications of this enterprise system allow for reusing existing concepts in its approach, the opportunity to consolidate and simplify letter generation and management across VBA, and the ability to plan for complexity.

Describe the problem in technical terms	We need an Enterprise level Line of Business agnostic tool to create and manage correspondence that can integrate with external API's without the need for a code change.
What are the architectural components in the problem's context?	New tenants: Resource Manager, Template Manager, Letter Manager, ECM-L Service.
What are the integration points that might be impacted by this problem?	Each of the new tenants will integrate with Line of Business specific API's on an as needed basis.
What technologies are involved?	Springboot, React, Kafka, iText, Lexical.
Is this related to already-stated technical debt?	No.
If the problem is with a technology, what were the reasons for using this technology in the first place?	N/A
Has an attempt been made before to fix the problem?	No. The existing components in the DocGen namespace were not designed to accomplish this goal / solve this problem and are too entwined with Line of Business specific logic to be repurposed to solve it.
Did it fail or did it just not finish?	N/A.
Who are the technical PoCs for that attempt?	N/A.

What components are impacted by this problem?	-All VA letters/correspondence -Core, Ratings, Awards
Are there current mitigations that help alleviate the impact of the problem?	No.
What are the technical dependencies for this? What does this depend on? What depends on it?	The new tenants proposed will stand on their own and external Line of Business specific API's can be configured in as dependencies as needed.
Based on the SV-1 and SV-9 generated what is estimated monthly cost for this new service? Upload estimate of resources from the AWS Pricing Calculator	TBD

Key Concepts and Summary

Mission Statement

Streamline correspondence management in an enterprise, application-agnostic way, to improve the speed and quality of the communications between the VA and the people it serves.

Goals of the ECM-L Ecosystem

- Reduce redundancy by maintaining all correspondence in a single system
- Empower the VA by giving them the tools to manage correspondence without developer intervention
- Allow for changes to correspondence to be published outside of a development cycle / release window
- Improve the quality of correspondence testing to ensure that our communications are always as accurate as possible
- Create a platform for integration that can fit seamlessly into existing application and Line of Business workflows
- Provide stable and reliable content for automated systems in an environment where that content is user maintained

The Building Blocks of Correspondence

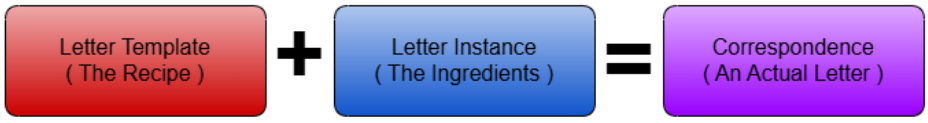
The Basics of Correspondence

To generate a piece of correspondence for a Veteran, what is needed?

First we need to know what that correspondence should look like. In the ECM-L ecosystem, we call this the ***Template***. The Template is similar to a *recipe* for a given piece of correspondence.

A recipe isn't enough though, we also need the *ingredients*. A Template may call for and need the name of a Veteran, or their periods of service. We collect these ingredients in what we call a ***Letter Instance*** - which represents an "instance" of a given Template.

If an example of a Letter Template might be called the "Development Letter", then an example of a ***Letter Instance*** would be John Smith's "Development Letter." This is a specific piece of correspondence, for a target audience, that is based on the recipe given to us by that Template.



We are able, in the end, to combine the "ingredients" we've collected in the Letter Instance, and the "recipe" contained in the Letter Template, to generate and send out our correspondence, so that John Smith's Development Letter contains all of the pertinent information that he needs.

Gathering the Ingredients

Sometimes a piece of correspondence requires information specific to the recipient, and there are different sources of that information. Depending on where the information is coming from, and where it "lives," we have a different means of gathering the information for our correspondence.

We can separate the information into two categories: Input-Driven Information and Resource-Driven Information.

Input-Driven Information (Provided by Users)

Any information "given" to the ECM-L components. This can be provided by a human being, via Template-driven dynamic forms in the Letter Manager, or by an automated system, via an API call.

All user-provided information is managed in a Template via what we call *Inputs.*

Information Provider	Gathering Method	Examples
Human User	Provided through the Letter Manager interface, by a human actor, using a Template-driven Dynamic Form, structured based on the needs of the individual Templates.	A Letter Manager user using the Dynamic Form in the Letter Manager to: ■ Select from a list of VA forms that should be included in the correspondence as Attachments. ■ Select from a list of evidences that should be referenced in the text of the correspondence. ■ Provide the total monthly income for a recipient.
System User	Provided by an automated or system actor via requests to the ECM-L components.	An automated system sending a request to the ECM-L API containing: ■ The periods of a Veteran's military service. ■ The nature of a Veteran's discharge. ■ Banking information, such as routing numbers and account types.

All user-provided information is managed in a Template via what we call *Inputs.*

Inputs are configured in a Template and can be used within the Template content, and these configurations determine the "shape" of the Letter Instances created using those Templates. When a user accesses the Letter Manager, the Dynamic Form that they use to fill in the missing content is based on the Inputs that were configured in the Template Manager.

Resource-Driven Information (Collected from External Services)

Any information that the ECM-L components needs to be configured to retrieve itself, by reaching out to external, Line of Business specific services.

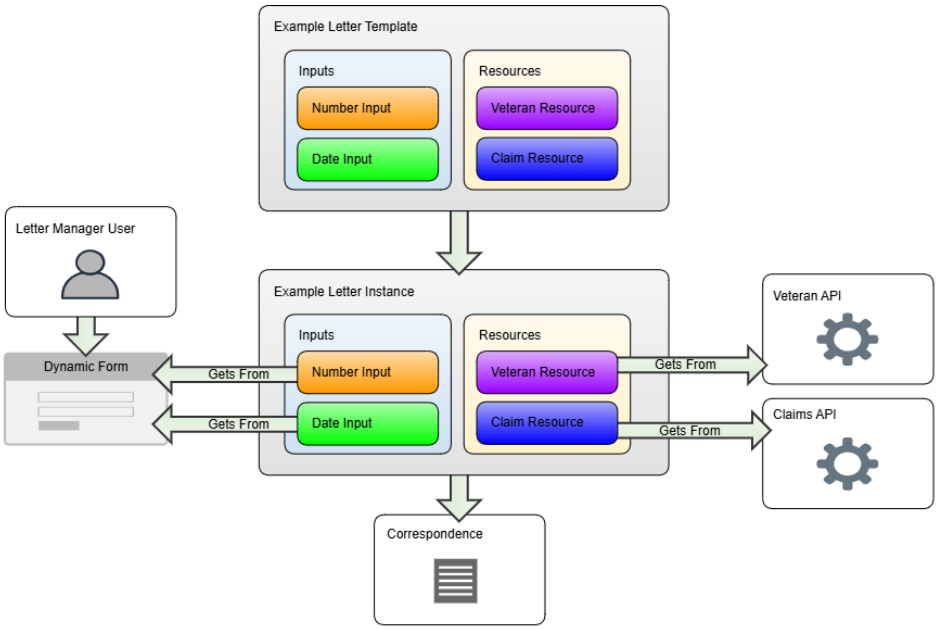
All user-provided information is managed in a Template via what we call *Resources.*

Information Provider	Gathering Method	Examples

Line of Business Service	The ECM-L components must reach out to and retrieve this data from the Line of Business's services on their own, using Connection Configurations and Resources pre-configured by a Resource Manager user.	The ECM-L components use a Resource and Connection Configuration set up by users in the Resource Manager to:
		• collect Veteran data from a Participant ID using the Veteran API.
		  ○ First and last name.
		  ○ Final disposition.
		  ○ Gender.
		• collect Claims information from a Claim ID using the Claims API.
		  ○ Claim Type.
		  ○ EP (End Product) Code.
		  ○ Jurisdiction.

Combining the Ingredients into Correspondence

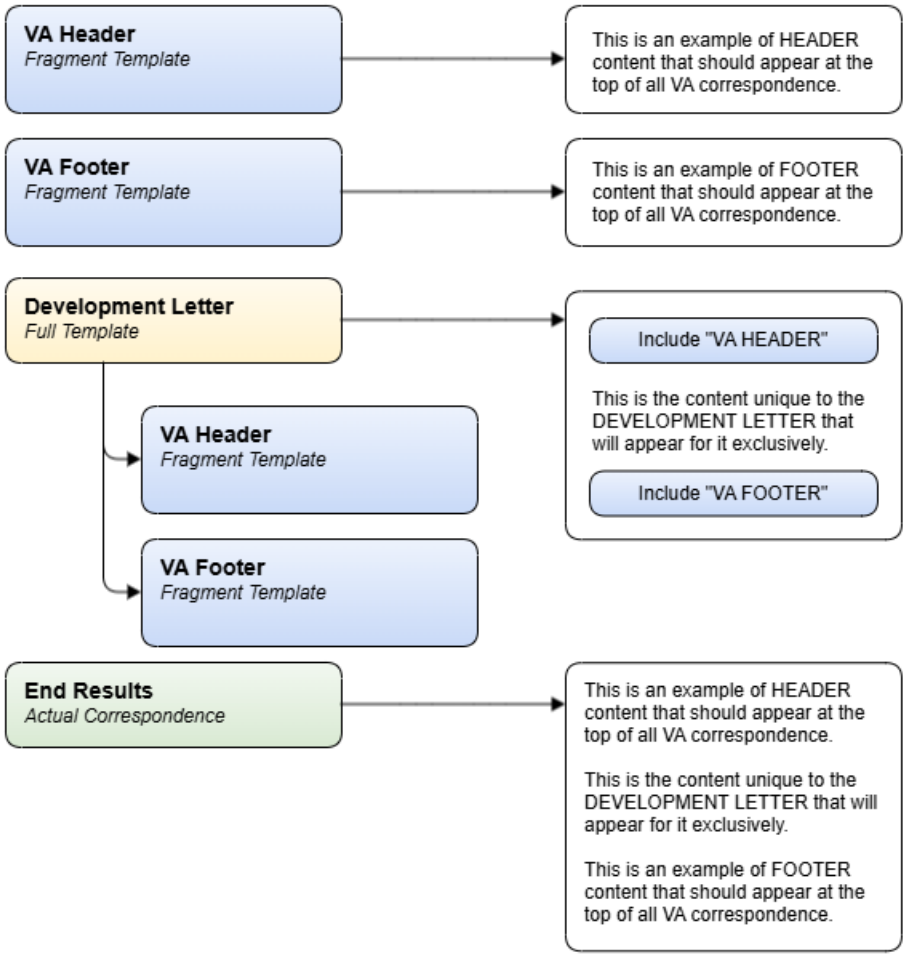
Once a Template is configured with the right Resources and the right Inputs the ECM-L eco-system can gather the information necessary to generate the correspondence.



Reusable Content

Most of our Lines of Business will have content such as a shared "header," or a shared "footer," that need to appear the same on all of their pieces correspondence.

ECM-L handles this need elegantly by introducing the concept of a ***Fragment Template***. Whereas a ***Full Template*** represents the full content and configuration of a specific piece of correspondence, Fragment Templates represent re-usable content that can be *included* inside of these Full Templates.



The Four Major ECM-L Components

The ECM-L ecosystem is made up of four sibling components that work together to provide Lines of Business all the tools and power they need to manage and maintain correspondence at a streamlined enterprise level. These components interact with each other via a robust "event" based architecture that allows for external systems to listen for and react to specific events to cover and address needs that are specific to their own use cases, such as the need to generate "Tracked Items" based on content in a Letter Instance as it is Finalized.

Name	Purpose
Resource Manager	Most of the correspondence we send out includes data that the Lines of Business have already collected. Things like the Veteran's name, their periods of service, their beneficiaries. The Resource Manager allows Lines of Business to create "Resources" that allow us to pull that information in to our generated correspondence.
Template Manager	This is where all of the different correspondence are managed by the individual Lines of Business. Using this tool they're able to update, test, schedule, and publish changes to their correspondence on their own time, using their own unique business processes.
Letter Manager	Much of the correspondence that gets sent out is user-driven. Users are identifying what type of correspondence should be sent, inputting data relevant to the correspondence, reviewing it, approving it, finalizing it, and sending it out to its intended audience. The Letter Manager is where all of that takes place.
ECM-L API	There are times when correspondence should be sent out without user interaction. Automated systems like the Pension Automation project or the Automated Document Generation (ADG) project will provide all of the information needed to generate the correspondence *instead* of a human user. The ECM-L API is the outward facing "gatekeeper" of the ECM-L eco-system, that these automated systems will communicate with. It's also where the PDF rendering engine lives, where we combine the fruits of the Template Manager and the Letter Manager into the actual correspondence.

Glossary of Terminology

Use this glossary as a guide as you navigate through these documents to better understand some of the core concepts and terms that we use in the ECM-L ecosystem.

Key Term	Description
Changeset	A collection of Template-related content that a user wants to publish at a specifically scheduled time. This can include new Template Versions, File Versions, or the retirement of one or more pieces of versioned content. This is the mechanism for managing all of the content within the Template Manager, including creating, testing and publishing, and retiring Template Versions and all other versioned content.
Connection Configuration	A collection of the Configuration information necessary for the ECM-L components to access another system's API so that it's Resources can be used by Templates and Letters.
Data Resource	Any information that we want to collect for a Template that will be provided by an external Line of Business specific system instead of being collected directly from a user. These are defined in the Resource Manager and configured by a Line of Business administrator, and the actual data is provided by an endpoint in a Connection Configuration.
Dynamic Form	The Template-driven form in the Letter Manager that users utilize to provide information needed for the pieces of correspondence they are working on. This same form is provided to Template Manager users when crafting Test Scenarios that are used to test their content before publication.
File	Any file-based assets that can be used by a Template such as a PDF or an Image. These live within the ECM-L eco-system and are managed within Changesets in the Template Manager like a Template, as opposed to File Resources which point to Files hosted externally.
File Resource	A type of Resource used to pull in different file-based assets for use in a Template, such as an Image or PDF.
File Version	A specific version of a File within the ECM-L.
Fragment Template	A Template that represents a piece of re-usable content that can be included inside of another Template. Fragment Templates are a useful way to handle things like common headers and footers. A Fragment Template cannot be instantiated as a Letter Instance on its own.
Full Template	The structured layout and all necessary components, including configuration information (e.g., resources, input types, conditions, restrictions, attachments, etc.) and content required to create a Letter Instance.
Image	A graphical File that can be inserted into Template content and rendered with a Letter Instance as a part of the PDF.
Label	Concise descriptive tags used by Changeset Editors to categorize and organize Templates.
Legacy Correspondence Management	All of the different legacy methods that the different Lines of Business used to manage their correspondence, that the ECM-L ecosystem seeks to replace, consolidate, and streamline.
Letter Instance	Letter Instances point to a specific Template Version and represent all of the information the Template requires for the generation of correspondence. ECM-L can return a Letter Instance as JSON data for situations where a Line of Business needs to evaluate that data or render it as a PDF.
Letter Instance Metadata	A subset of the data related to a Letter Instance that helps a user differentiate it from other Letter Instances, but that does not contain any data collected from Resources or users, and can not on its own produce a generated PDF. Contains information such as the name of the Template that Letter Instance was created for, and the date was created.
Line of Business	An organizational segment of operational and strategic activities. Each of the Templates and Changesets belong to a Line of Business.
Major Version	The Major Version of a Template is the number that precedes the decimal. For Version "3.2", the Major Version is "3".
Minor Version	The Minor Version of a Template is the number that follows the decimal. For Version "3.2", the Minor Version is "2". Any new Version of a Template that only updates the Template's Content, becomes a new Minor Version of the Template.
PDF	Portable Document File, a type of File that can be used by a Template as an Attachment.

Resource	Content that needs to be retrieved from a Connection Configuration, which allows the ECM-L components to communicate with outside systems. There are Data Resources which cover any data that might be returned by one of these systems and File Resources that represent file-based assets.
Retirement Action	Retirement Actions point to versioned content such as a File Version or a Template Version and upon publication will retire that specific version of that content. (See Retirement)
Template	A versioned representation of a type of correspondence, potentially containing reusable content such as Fragment Templates and contextual data provided by Data Resource placeholders, necessary to successfully instantiate and render a Letter Instance. Templates belong to a specific Line of Business. There are two types of Templates - Full Templates and Fragment Templates.
Template Version	A specific version of a Template, composed of a combination of a Major and Minor Version number. If the Major Version of a Template is "3", and it's Minor Version is "2", then the Version is "3.2"., composed of a combination of a Major and Minor Version number.
Terminal State	A state that an element of the ECM-L system can enter into it that indicates user activity for that element has ended, such as a Letter Instance entering into the DELETED or CANCELLED or FINALIZED state, from which it can not return.
Test Scenarios	Test Scenarios are a way for Template Manager users to create sets of sample data that can be used to create example PDF's using the Templates they've created so they can confirm they are working as expected before publication. Users can add any number of Test Scenarios to a Template Version in a Changeset to test out the different permutations of the content based on different example data.
Workflow	The LoB-specific business process context in which the Letter Manager MFE operates by organizing and categorizing Full Templates and associating specific Key Parameters required for letter generation. An example of a workflow might be, "Developing a C&P Claim," or "Rating a Disability." It describes what we're doing and the purpose of the related correspondence generation.

Other Important Concepts

Letter Instance Archival

Why do we need Letter Instance archival?

This tiered storage approach optimizes costs by matching storage expenses to access patterns - keeping frequently accessed letter data in fast, moderately-priced S3 storage while moving rarely accessed historical records to cost-effective Glacier archiving, while maintaining compliance with VA record retention policies and keeping the transactional database lean and performant.

How long before a Letter Instance is archived?

Letter Instances will be archived based on the Line of Business configuration - by default, it will transition into Archived status after 60 days. Lines of Business can request that this default configuration be overridden to a custom number of days instead.

How do we handle Letter Instance archival?

Letter Instances that have entered a Terminal State will be stored in an S3 bucket for immediate access during the active period when reprints, disputes, or audits are most likely, and will then be automatically migrated to Amazon Glacier storage after a configurable retention period that varies by Line of Business based on their specific regulatory and operational requirements. This configuration of the retention period takes place during the Line of Business intake process and updating the configuration is managed through a Change Request.

The Letter Instance Metadata will remain and users within the Letter Manager will be able to see that the Letter Instance exists and was moved into the ARCHIVED state, which will allow them to file a ticket, if necessary, to move the rest of the Letter Instance data back from Glacier so that it can be accessed again.

Retirement

Why do we need the concept of Retirement in the ECM-L?

The idea behind Retirement is that different Lines of Business will have different business processes that outline how a specific piece of correspondence, such as a Letter Template, might be discontinued. Based on the need for an enterprise level solution that is business agnostic, we've adopted a Retirement philosophy that allows for these LOB's to come up with a plan to discontinue and / or replace a Template on its own time, using its own business processes.

When a LOB decides they no longer want a specific version of a Template to go out to its audience, they can have their users go in and Retire that Template Version. Now that it has been Retired, the Line of Business can rest assured that no new Letter Instances will be created based on that Template – it has in effect been turned "off" and is no longer available for use.

What happens when something gets Retired?

Once an element of the ECM-L ecosystem has been *Retired*, it can no longer be used for anything ***new***.

It no longer appears for selection in any lists that would include it as an option, and no new items can be created using that item – however, any pre-existing configurations or places that already made use of the retired item will continue to function just as before. This ensures safety and continuity for current configurations while preventing the use of outdated or deprecated options moving forward.

Item	Retirement Impact
File Version	New Template Versions can no longer be configured to use this File Version - it is no longer available for use inside of the content of a Template in the case of Images, or as an attachment in the case of PDF's. Existing Template Versions already configured with this File Version can continue to use it as before.
Key Parameter	New Resource Versions and Workflows can no longer be configured to use this Key Parameter - it does not appear in the list of available Key Parameters. Existing Resources and Workflows already configured with this Key Parameter can continue to use it as before.
Resource Version	New Template Versions can no longer be configured to use this Resource Version - it does not appear in the list of available Resources. Existing Template Versions already configured with this Resource Version can continue to use it as before.
Template Version	New Letter Instances can no longer be instantiated using this Template Version. Existing Letter Instances instantiated from this Template prior to its Retirement can still be worked and viewed as before.
Workflow	New Letter Instances can no longer be created within this Workflow. Existing Letter Instances created before the Workflow was Retired can still be worked and viewed as before.
Connection Configuration	New Resource Versions can no longer be configured to use this Connection Configuration - it does not appear in the list of available Connection Configuration(s). Existing Resource Versions already configured with this Connection Configuration can continue to use it as before.

How does ECM-L handle Template Versions with Retired components / dependencies within a Changeset?

A Template Version cannot be marked as Completed while it contains Retired components / dependencies.

After completion, the Template Version has moved passed the gate, so to speak. If after it moves through that gate, into the Completed status, something it uses is Retired, we will allow the Line of Business to continue to move it onwards through to Publication, but will alert those viewing the Changeset that it's using Retired components along the way.

This way the Line of Business can decide if they want to send the Template Version back to Draft so that those Retired components can be removed, or move forward with the Publication anyway.

While a Template Manager user is in a screen where they are Testing, Approving, or Scheduling a Template Version for Publication, a warning banner will appear if it is using Retired Components.

Versioning

Versioning of Material

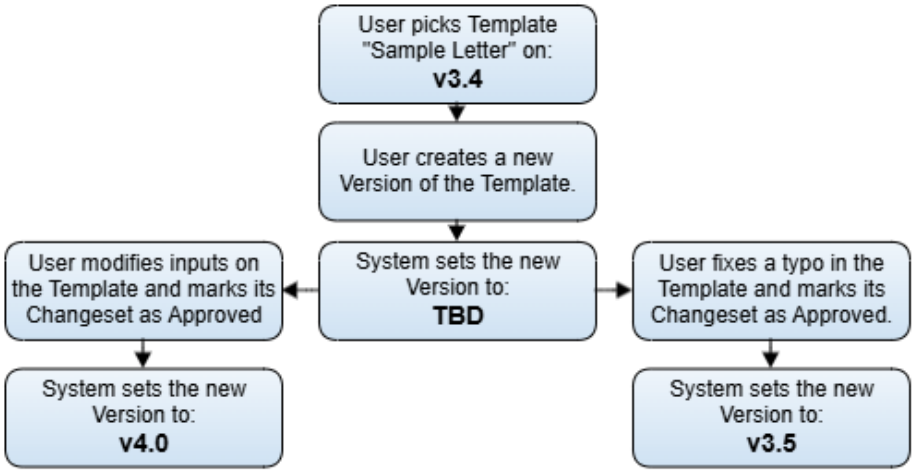
Versioning the material managed in the Template Manager allows us to see exactly which ingredients were combined to create a given piece of correspondence. By maintaining a running historical record of all of the versions of these materials we can recreate correspondence today exactly as it was created yesterday, allowing us to compare changes against old versions and audit historical data when necessary.

Major and Minor Template Versions

When a user is modifying a Template, there are certain non-breaking changes that are superficial and cosmetic, such as fixing a typo, and there are deeper changes, breaking changes, that alter the shape of the Letter Data in ways that could cause problems for downstream systems.

We help to mitigate this and identify when a user is making a breaking or a non-breaking change. For breaking changes, the new version of the Template is given a new Major Version. For non-breaking changes, the new version of the Template is given a new Minor Version instead.

Prior to the Changeset being marked as Approved, the version number for the Template Version will be set to and rendered as "TBD" (To Be Determined).



Template Version Duplication and "Restoring" Retired or cancelled Template Versions

When a user is creating a new Template or Template Version they're able to choose an existing Template Version to base it on – giving them the ability to duplicate Template content, restore Retired content, and to improve the speed at which they can manage and deliver new correspondence.

> ⚠ Certain content may not be able to copy over into the new Template Version – for example, if the version we want to copy from contains a Retired Data Resource, we will not be able to pull that Data Resource in to our new Version, and any usages of the Retired Data Resource may need to be cleaned up by the Template Editor.

New Template Versions

Users in the Template Manager are able, when creating a new Template Version, to base that new Version on any of the pre-existing Versions of the Template that they so choose. The user is even able to choose Versions that had been Retired.

By default, the new Template Version will be based on the last Version that was published, but if they wanted to create a new Version based on a Version that pre-dates that, the user can choose to do that.

Completely New Templates

They are also, when creating a completely new Template, able to base that on a Version of an already existing Template – which gives users the power to make copies of Templates, or base a new Template on one that already exists.

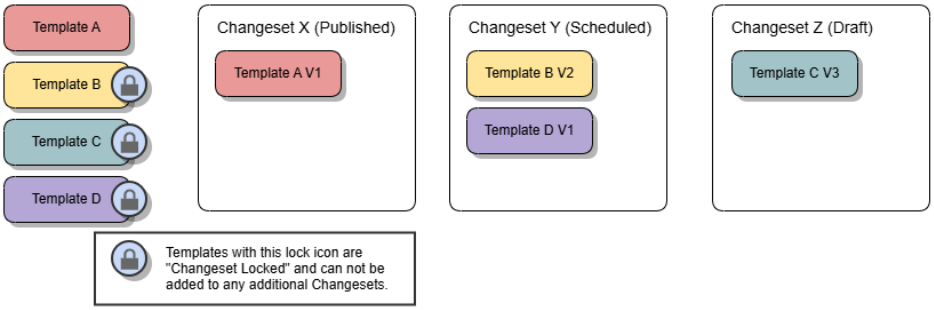
Changeset Locking

Templates that are actively being worked on in a Changeset are essentially "Locked" and cannot be used in other Changesets.

When a Changeset is in a Non-Terminal State, the Templates in that Changeset are "Locked" and can only exist in that single Changeset and cannot be included in any new Changesets until that Changeset is either cancelled or Published.

"Terminal" Changeset States:

- cancelled
- Published



Workflows

What is a Workflow?

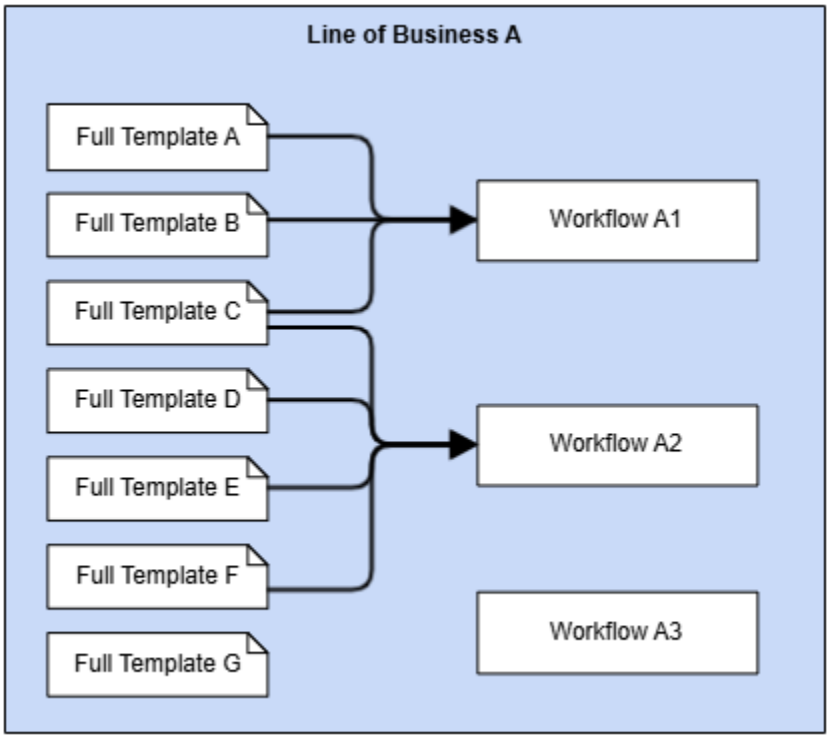
A Workflow answers the question: Why are we sending out this correspondence? The user can indicate that they are "Developing a Claim," or "Rating a Claim," or "Awarding a Claim," and we can cater the Templates available to them based on that chosen Workflow.

These relationships between Full Templates and Workflows are managed in the Template Manager so that ECM-L users are empowered to control those relationships on their own.

When a Letter Manager user goes to create a piece of correspondence, the first thing they need to do is choose from a list of optional Templates to base that Correspondence on - the Workflow they are in will help to shape that list of options, so that they're only shown options that are relevant to what they are currently trying to accomplish.

Example

This diagram illustrates the relationship between Workflows and Full Templates within a Line of Business.



Full Templates A, B, and C, are aligned to Workflow A1, so a user accessing the Letter Manager through that Workflow will be able to choose from those Templates when creating a Letter Instance. Even though Full Template D is a part of the same Line of Business, since it is not aligned to Workflow A1, it will not appear in the list of available Templates for that Workflow.

Line of Business	Workflow	Available Templates
A	A1	A, B, C
	A2	C, D, E, F
	A3	None

System Actors and Automated Activities

The ECM-L components provide the perfect platform for Lines of Business to integrate with for any sort of automated letter generation needs.

Automatic Letter Generation

The ECM-L API provides an interface that your systems can use to interact with the ECM-L components as if they were a human end-user. Systems are able to Create a Letter Instance, and Finalize a Letter Instance, and even to run a Letter Instance all the way through its lifecycle from Creation to Finalization in a single request.

Auto-Finalize / Fast-Tracking

System Actors can use the ECM-L API's "createLetterInstance" with the "autoFinalize" flag set to true in the request to in one action create a Letter Instance and move it all the way through to Finalization. This is for Systems that want to create a Letter Instance and get a PDF without user interaction.

They would first hit this "createLetterInstance" endpoint which would result in a Finalized Letter Instance, and then they would hit the ECM-L endpoint to Render the Letter Instance as a PDF ("renderLetterInstance").

Other System Actor Activities

An external system may want to take other actions within the ECM-L components based on user interaction in their own system, or on something else like a rules engine, and all of that is possible via the ECM-L API as well. All of the operations in the ECM-L API are available for use by System Actors.

Eventing

The ECM-L components emit Kafka Events as actions are taken throughout the system. When a Letter Instance is Finalized, for example, an event is emitted that indicates this has happened. A Line of Business system can be updated to listen for those Finalization Events and to take actions that will be triggered when those Events take place.

Because we've provided these enterprise tools for the Lines of Business to handle these things on their own, specialized LOB-specific behavior no longer constitutes development work and code changes within the ECM-L components, and all of these changes occur within those external components instead.

Eventing Use Case Example

A Line of Business needs to create something called a "Tracked Item" based on the Letter Data that was entered in by its Letter Manager end-users, anytime a Letter is Finalized. They can update their system to listen for Letter Finalization Events and have their system hit the ECM-L API's "getLetterInstance" endpoint to get the Letter Instance data. Their system can now get the information from the Letter Instance data and use that to create their Tracked Items.

Examples of Events

The Eventing framework is robust and we essentially produce an Event for any life-cycle change for any element within the ECM-L components. This allows for Lines of Business to automated specialized activities based on any changes to their Templates, Letters, and other content.

The Lifecycle Diagrams elaborate on the Element-specific Events that exist, but here are some examples:

- Connection Configuration Published Event
- Key Parameter Type Retired Event
- Changeset Submitted for Approval Event
- Template Version Retired Event
- Letter Instance Finalized Event

Style Guide and Special Characters

Style Guide

The underlying stylesheets and the styling / formatting options provided to Template Manager users to create documents generated by the ECM-L components are based on the VA Style Guide. There is no way to update or deviate from the provided styling options without an enhancement and development time to address it. For more information about the VA Style Guide, see here: 2019 VA Style Guide.

Special Characters

The content of Correspondence can be complex and can require special characters that other more normalized data inputs allow. For the ECM-L components we have created a whitelist of special characters that are allowed and have outlined situations in which those allowable characters might change or vary. Any updates to this list of acceptable special characters will require an enhancement and development time to address.

Navigate here to view the details about which Special Characters are allowed: Special Character Allowances in the ECM-L Components.

JSON Path Expression Examples

He is a list of some example JSON Path expressions, which our users would use during the configuration process for a Data Resource, for example.

Expression	Description	Example JSON	Result
`$.veteranName`	Simple property access	`{"veteranName": "John Smith", "serviceNumber": "123456789"}`	`"John Smith"`
`$.address.city`	Nested property access	`{"address": {"city": "Richmond", "state": "VA", "zip": "23230"}}`	`"Richmond"`

`$.claims[0]`	Array element by index	`{"claims": ["disability", "education", "healthcare"]}`	`"disability"`
`$.claims[*]`	All array elements	`{"claims": ["disability", "education", "healthcare"]}`	`["disability", "education", "healthcare"]`
`$.claims[-1]`	Last array element	`{"claims": ["disability", "education", "healthcare"]}`	`"healthcare"`
`$.veterans[*].name`	Property from all array objects	`{"veterans": [{"name": "John Smith"}, {"name": "Jane Doe"}]}`	`["John Smith", "Jane Doe"]`
`$..benefitType`	Recursive descent (all benefitType properties)	`{"veteran": {"benefitType": "disability"}, "spouse": {"benefitType": "survivor"}}`	`["disability", "survivor"]`
`$.veterans[?(@.serviceYears > 10)]`	Filter array by condition	`{"veterans": [{"name": "John", "serviceYears": 15}, {"name": "Jane", "serviceYears": 5}]}`	`[{"name": "John", "serviceYears": 15}]`
`$.claims[?(@.amount > 1000 && @.status == 'approved')]`	Multiple filter conditions	`{"claims": [{"amount": 1500, "status": "approved"}, {"amount": 500, "status": "pending"}]}`	`[{"amount": 1500, "status": "approved"}]`
`$.benefits[?(@.type in ['disability', 'education'])]`	Filter with IN operator	`{"benefits": [{"id": 1, "type": "disability"}, {"id": 2, "type": "healthcare"}]}`	`[{"id": 1, "type": "disability"}]`
`$.veterans[?(@.awards[*] == 'Purple Heart')]`	Filter by array content	`{"veterans": [{"awards": ["Purple Heart", "Bronze Star"]}, {"awards": ["Good Conduct"]}]}`	`[{"awards": ["Purple Heart", "Bronze Star"]}]`
`$.veterans[0:2]`	Array slice (first 2 veterans)	`{"veterans": ["Smith, John", "Doe, Jane", "Johnson, Bob", "Williams, Sue"]}`	`["Smith, John", "Doe, Jane"]`
`$.lineOfBusiness[?(@.name && @.id)]`	Filter objects with required fields	`{"lineOfBusiness": [{"name": "Claims Processing", "id": "VBA-001"}, {"name": "Benefits Administration"}]}`	`[{"name": "Claims Processing", "id": "VBA-001"}]`

Date Formatting

Throughout BIP and the ECM-L components the dates are passed around and stored in UTC format using *ZonedDateTime* to provide a single global reference point, ensure all timestamps are in the same timezone to provide reliable chronological ordering and records, and that this time zone is communicated properly to the consumers of our API's.

In the UI's, and in the generated PDF's, when these dates are presented to an end user, they are displayed for the "ET" time zone, also known as the "America / New York" time zone, to comply with VA standards and guidelines.

History and Audit Trail Tracking

Comprehensive timestamped history tracking across all three management systems allows us to maintain an audit trail of user activities and system changes that take place in the ECM-L components. In the Template Manager, the system maintains a historical record of all the Changesets and versioned content those Changesets manage, with a historical view that tracks their movement through their various lifecycles. Similarly the Resource Manager and Letter manager maintain a timestamped historical record of the content they manage.

These historical records outline when a change was made, a description of the change, and who made it, creating a comprehensive audit trail that supports operational troubleshooting and regulatory compliance requirements.

System Level Key Parameter Types

While our goal is to keep the ECM-L components business level logic and concept agnostic, allowing for individual Lines of Business to define the shape and content of their correspondence and the resources those pieces of correspondence use, to provide certain pieces of functionality we are required to introduce certain System Level concepts that are shared across the VA.

Changes to the VA's use of these Key Parameter Types could require development intervention, since there are ECM-L components designed specifically to work with them.

System Level Key Parameter Type	Reason for Inclusion
Participant ID	The Participant ID is a common thread across most of the Lines of Business and more importantly is required to support Claim Evidence and Package Manager functionality.

New Name Uniqueness

To prevent ambiguity, confusion, and avoid conflicts, we enforce name uniqueness restrictions throughout the ECM-L components. In a general sense, names for things must be unique.

These restrictions apply whenever a new name is being decided, through the creation of something new, or through the renaming of something that already exists.

New Name Type	New Name Uniqueness Restriction Details
Template Version	Must not match any current Template Name or ACTIVE / DRAFT status Template Version Name within the same Line of Business.
Resource	Must not match any existing Resource Name within the same Line of Business.
Input	Must not match any existing Input Name that is a sibling of this input. So if at the root level of a Template, there must be no other Inputs with that name at the root level, or if in an Input Group, there must be no other Inputs with that name in the Input Group, etc.
Workflow	Must not match any existing Workflow Name within the same Line of Business.
Key Parameter Type	Must not match any existing System Level Key Parameter Typeor any Key Parameter Type Name within the same Line of Business.

Solution Space and Discussion

The solution space describes the architectural context related to the problem/solution in a 4+1 style document. A 4+1 Design is a human-readable, standardized way of viewing an architecture from multiple perspectives. Each perspective is complementary, offering a different way to look at a particular architecture. Taken as a whole, a 4+1 Design is a comprehensive way to communicate an architecture.

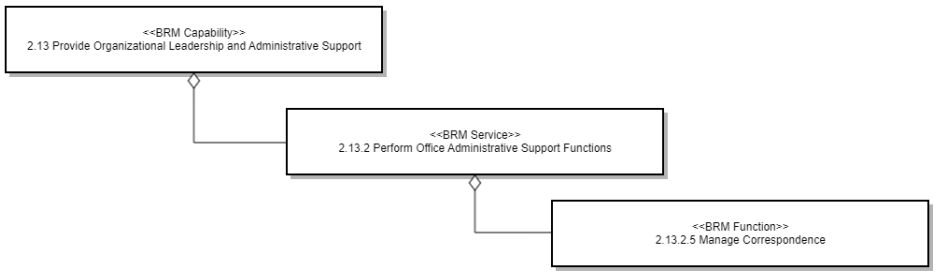
Use Case View

Capability Viewpoint Vision (CV-1) (VASI Description)

Enterprise Correspondence Management - Letters (ECM-L) will reside on the Benefits Integration Platform as a Service (#2295) and is under the Veterans Benefits Management System (VBMS #1728) Product. ECM-L is an enterprise-level system that streamlines the process of managing letter templates and generating letters across the Veterans Benefits Administration (VBA). It is accessible to multiple Lines of Business (LoBs) and allows each one to independently manage their own templates and generate correspondence from drafts through finalization. ECM-L can be integrated into the business logic of each LoB's systems through APIs, predefined hooks, and other integration points, as its ecosystem is LoB-agnostic and highly extensible.

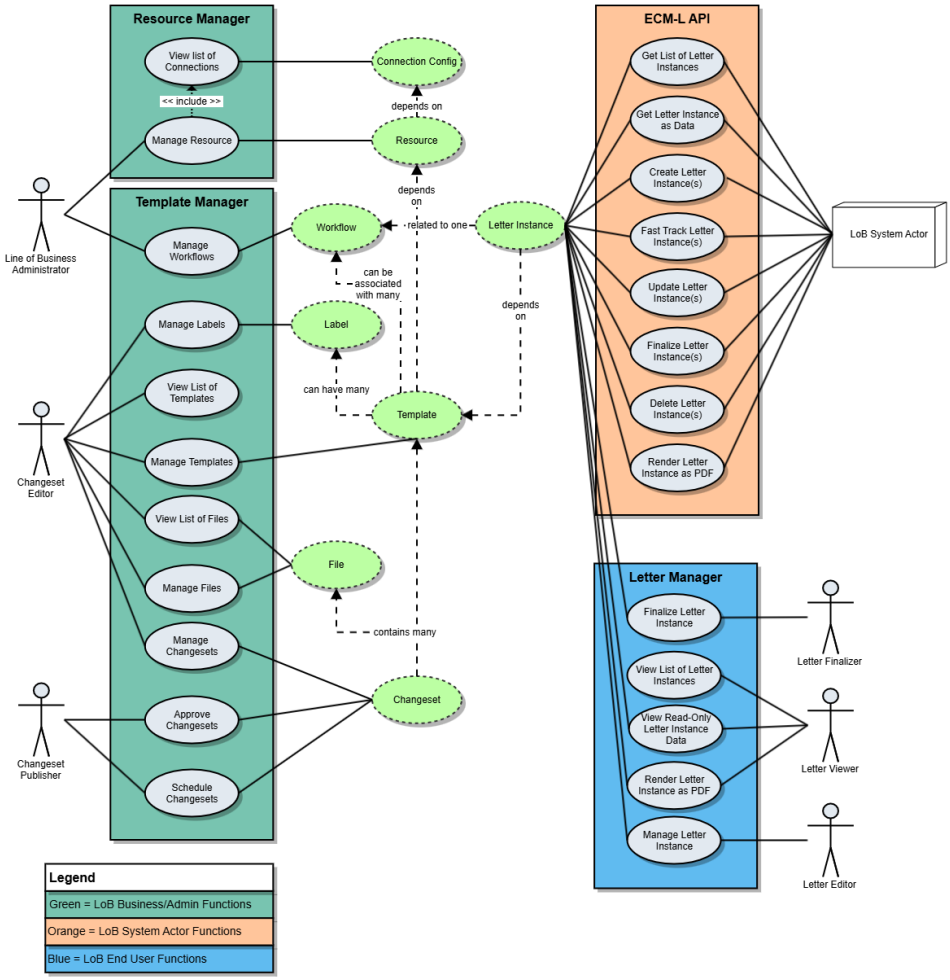
Capability Viewpoint Taxonomy (CV-2)

The following diagram provides an architectural description of the overall, high-level vision and for the capability of this component/tenant.



Use Case

The following diagram visually represents the interactions between users, or actors, and the system. The diagram illustrates the functional requirements of the system.



Main Actors

The following table describes the actors listed in the diagram above. An actor is user in the scope of this architecture and can be a human user or a "system actor" as appropriate.

Name	System	Description
Line of Business Administrator	Resource Manager	A User with the requisite roles and permissions to manage Resources and Connection Configurations.
Line of Business Administrator	Template Manager	A User that manages Workflows, including creating, modifying, and retiring Workflows, configuring them with Key Parameters, and aligning Full Templates to them.
Changeset Editor	Template Manager	A User that manages Templates through Changesets, including creating, updating/modifying, testing, recalling, and cancelling Changesets, to facilitate the publication and/or retirement of one or more Template Versions. This user can interact with and manage Template Versions, File Versions and Retirement Actions in a Changeset, allowing for them to create (specifying a Template's content and configuration), read, update, test, cancel, and retire specific Template Versions. This user can also add Labels to Templates.
Changeset Publisher	Template Manager	A User responsible for Approving, Scheduling, and Publishing Changesets. Changesets can contain Template Versions, File Versions and Retirement Actions. When a Changeset Publisher Approves a Changeset all contents are APPROVED.

Line of Business System Actor	ECM-L API	External systems that interact with the ECM-L ecosystem to conduct automated operations such as sending out correspondence without user intervention.
Letter Finalizer	Letter Manager	A user with the ability to Finalize Letter Instances in the Letter Manager.
Letter Editor	Letter Manager	A user with the ability to create, read, update, and delete Letter Instances in the Letter Manager.
Letter Viewer	Letter Manager	A user with the ability to view list of Letter Instances, Letter Instance Data, and the Rendered PDF in the Letter Manager.

System/Tenant/Application Description

The following table describes each system/tenant/application (rectangles) listed in the diagram above.

System Name	Description
Resource Manager	The tool that allows users to create and manage Resources and Connection Configurations.
Template Manager	The tool used to manage Changesets, Workflows, Labels and Templates.
ECM-L API	The ECM-L API is the outward face of the ECM-L system and allows for external system actors to interact with the ECM-L system in lieu of a human actor directly interacting with one of the individual components.
Letter Manager	The tool used to manage Letter Instances.

Main User Functions (Use Cases)

The following table describes each of the use cases (ovals) in the diagram above.

Actor(s)	System	Ability	Details
Line of Business Administrator	Resource Manager	Manage Resource	The Line of Business Administrator is able to create, modify, retire, and delete Resources (Data/File). The Line of Business Administrator will need to be able to view the list of AVAILABLE Connection Configurations to create Resources.
	Template Manager	Manage Workflows	The LOB Admin accesses the Template Manager system and is able to create, modify, and retire Workflows, and add Workflow Key Parameters as well as align Full Templates to those Workflows.
Changeset Editor	Template Manager	Manage Templates	The Changeset Editor accesses the Template Manager to create, modify, and retire Templates. This includes the ability to add or remove Resources for a Template. The Changeset Editor also has the ability to add Inputs to a Template and format the values, as well as add conditional Restrictions to a Template.
		View List of Templates	The Changeset Editor is able to get a full list of all Templates and their respective Versions, for the Line of Business they are working in.
		Manage Files	The Changeset Editor is able to manage, via a Changeset, different types of Files, such as PDFs that can be used as an Attachment and Images, that could be inserted into the content of a Template.
		View List of Files	The Changeset Editor is able to get a full list of all Files and their respective Versions, for the Line of Business they are working in.
		Manage Labels	The Changeset Editor accesses the Template Manager system and is able to create, modify, retire, and delete Template Labels, and add Labels to Templates.
		Manage Changesets	A Changeset Editor is able to create, test, and cancel Changesets.
Changeset Publisher	Template Manager	Approve Changesets	Once a Changeset is in "READY FOR APPROVAL" state, a user with the Changeset Publisher role can Approve the Changeset. Changesets can contain Template Versions, File Versions and Retirement Actions. When a Changeset Publisher Approves a Changeset all contents are APPROVED.
		Schedule Changesets	The Changeset Publisher is able to Schedule Publication of the Changeset immediately or in the future.
Line of Business System Actor	ECM-L API	Get List of Letter Instances	A Line of Business System Actor is able to get Letter Instances based on a given Workflow and Key Parameters.

		Get Letter Instance as Data	A Line of Business System Actor is able to get the Letter Instance as a JSON object.
		Create Letter Instance(s)	A Line of Business System Actor is able to create a Letter Instance.
		Fast Track Letter Instance(s)	A Line of Business System Actor creates a Letter Instance and "fast tracks" it through the Letter Instance lifecycle, through Finalization. This makes use of the same operation as the "Create Letter Instance(s)" use case but passing in an optional "autoFinalize" flag.
		Update Letter Instance(s)	A Line of Business System Actor has the ability to make modifications to a Letter Instance(s).
		Finalize Letter Instance(s)	A Line of Business System Actor has the ability to Finalize a Letter Instance(s).
		Delete Letter Instance(s)	A Line of Business System Actor has the ability to soft delete a Letter Instance(s).
		Render Letter Instance(s) as PDF	A Line of Business System Actor is able to Render a Letter Instance as a PDF.
Letter Finalizer	Letter Manager	Finalize Letter Instance(s)	The Letter Finalizer has the ability to Finalize a Letter Instance. Finalizing the Letter freezes its content and makes it immutable / unchanging.
Letter Viewer	Letter Manager	View Letter Instance(s) Data	The Letter Viewer has the ability to view Read-Only Letter Instance Data.
		Render Letter Instance(s) as PDF	The Letter Viewer is able to Render a Letter Instance as a PDF.
		View List of Letter Instance(s)	The Letter Viewer is able to view a List of available Letter Instances.
Letter Editor	Letter Manager	Manage Letter Instance(s)	The Letter Editor has the ability to create, read, update, and delete a Letter Instance.

Primary Systems High Level Operational Viewpoint (OV-1)

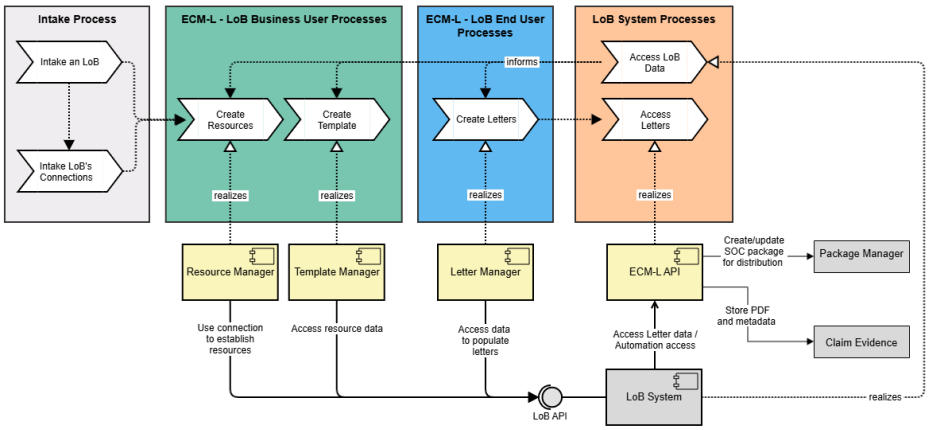
The diagram below is a high level overview of the main architectural components of ECM-L and the functions they facilitate in the value stream.

The first section of the value stream, the intake process, starts as a Jira ticket. This section consists of mainly manual processes provisioning space and storage, establishing user access requirements, and integrating with external systems. At this point in the process Line of Business (LoB) data needs will be identified and API connections will be established to accommodate those needs.

The ECM-L LoB Business User Processes box consists of two main values; creating Resources and creating Templates. Resources are assets that will point to LoB Data using the established API connections to populate Letter Instances as needed. The ability to create, test, and manage Resources is realized by the Resource Manager. Creating and managing Templates, or Letter Templates, is realized through the Template Manager.

The ECM-L LoB End User Processes box represents end user interactions with the system to instantiate Letter Instances generated from Templates and populated by user Inputs and Resources.

LoB System Processes encapsulates the ability to access, store, and send Letters as the LoB sees fit. Access to the Letters is capable through the ECM-L API.



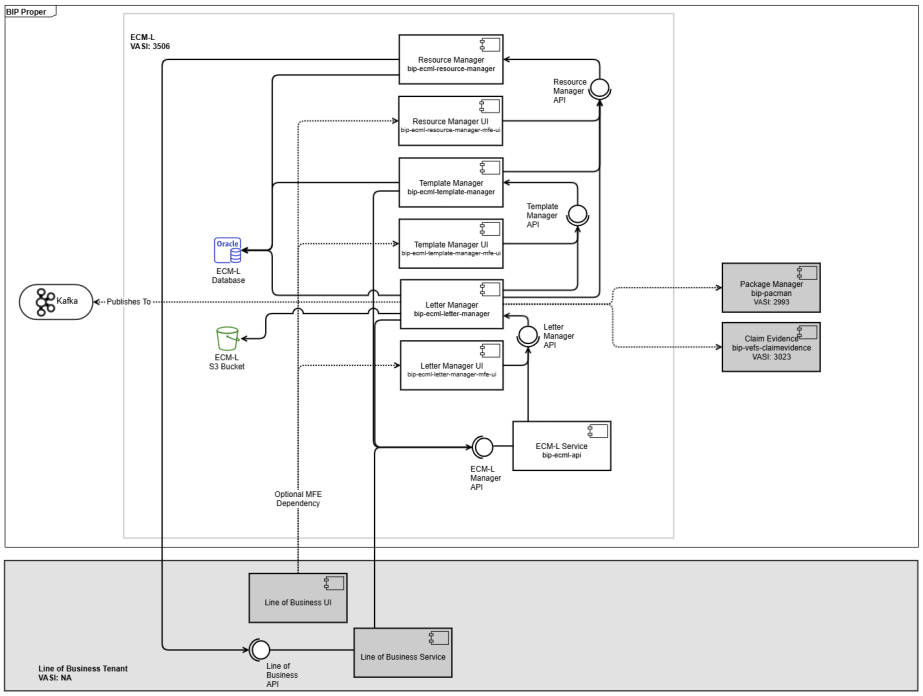
System (VASI)	System Item	Interaction
ECM-L (3000)	Resource Manager	Uses the LoB connection to create and manage Resources.
ECM-L (3000)	Template Manager	Creates and Manages the LoB Templates.
ECM-L (3000)	Letter Manager	Creates and Manages the LoB Letter Instances.
ECM-L (3000)	ECM-L API	Accesses the LoB Data and Letters for the LoB. Letter Instances are able to be stored in Claim Evidence and distributed via Package Manager.
VEFS (3023)	Claim Evidence	Stores the Finalized PDF related to a claim.
PacMan (2993)	Package Manager	Creates a package for mail distribution.
LoB System	LoB System	Uses its connection to establish Resources, Templates and populate Letter Instances with the appropriate data and access the Finalized Letter Instances.

Logical View

The Logical View describes the conceptual view of the functionality and of the systems that will participate. This view defines what nouns are involved and the relationship between them. For example, it may decompose the design into sub-systems and packages, and for significant packages, the decomposition into classes and class utilities. This section may include the Business Object Model, states and transitions for key objects, and other key details about classes and attributes.

Systems Interface Description (SV-1)

The following diagram provides a detailed view of system interfaces, depicting how systems and components interact with each other within the architecture.



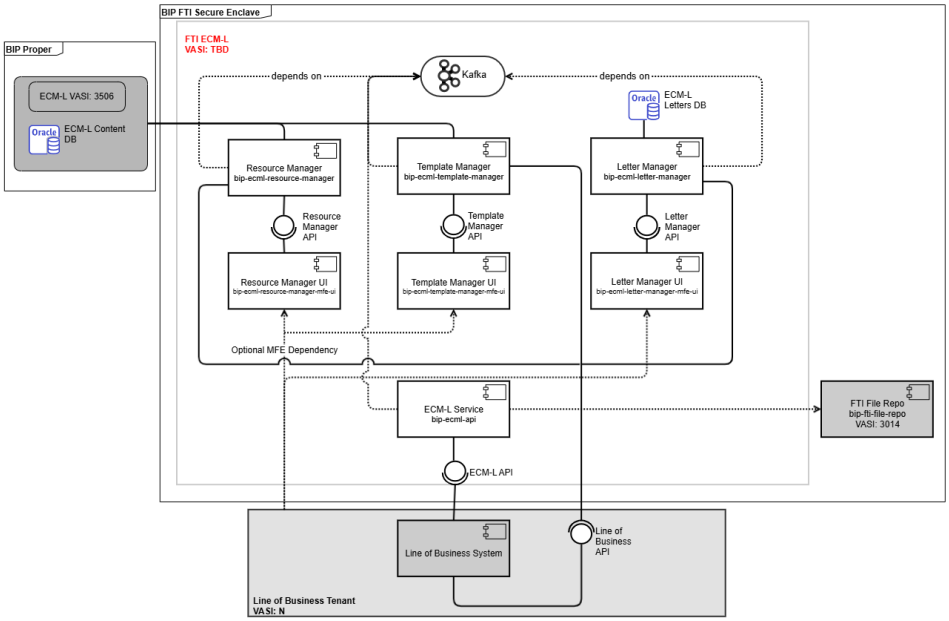
Legend
Gray Rectangle = External to core system(s)
White = Internal to core system(s)

Component References Table

Name	Description	Port	Protocol
ECM-L Content Database	Database containing all ECM-L related data. Database containing "dry" non-contextual data, uncontaminated with PII /PHI/etc.	1521	TCP
ECM-L Letters Database	Database for storage of contextual data containing PII/PHI/etc.	1521	TCP
Resource Manager API	Internal API for the Resource Manager	443	HTTPS /REST
Template Manager API	Internal API for the Template Manager	443	HTTPS /REST
Letter Manager API	Internal API for the Letter Manager	443	HTTPS /REST
ECM-L API	The API for the umbrella service for the ECM-L system, integrated with all of the internal components.	443	HTTPS /REST
Claim Evidence	External API for storage of documents in a Veteran's EFolder.	443	HTTPS /REST
Package Manager	External API for distributing packages of documents from a Veteran's EFolder.	443	HTTPS /REST

FTI Environment Example SV-1

This diagram provides a detailed view of system interfaces with service providers and service consumers, depicting how ECM-L components interact with each other in BIP FTI Secure Enclave for creation and management of FTI-related correspondence. The FTI Secure Enclave is a protected environment within Veterans Benefits platform specifically designed to handle Federal Tax information in compliance with IRS security requirements.



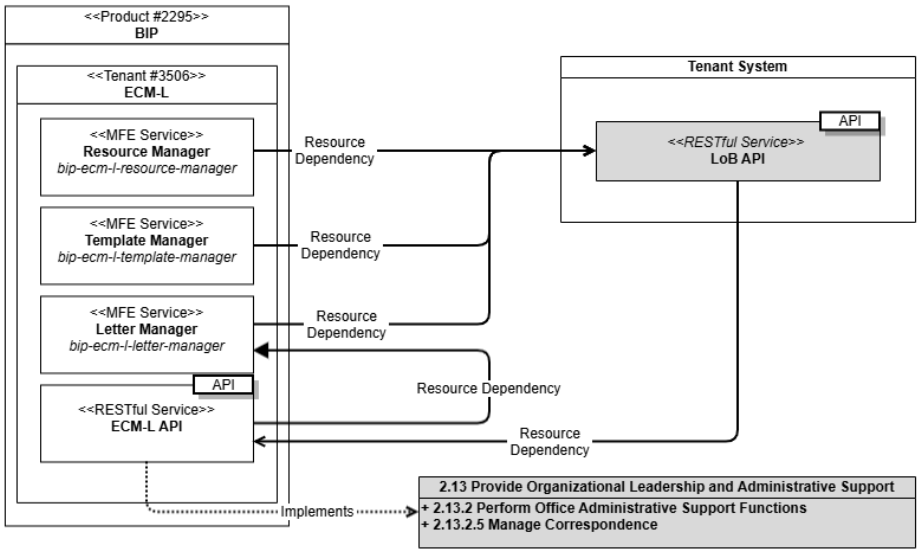
Legend
Gray Rectangle = External to core system(s)
White = Internal to core system(s)

Component References Table

Name	Description	Port	Protocol
ECM-L Content Database	Database containing "dry" non-contextual data, uncontaminated with PII/PHI/etc.	1521	TCP
ECM-L Letters Database	Database containing contextual data contaminated with PII/PHI/etc.	1521	TCP
Resource Manager API	Internal API for the Resource Manager	443	HTTPS /REST
Template Manager API	Internal API for the Template Manager	443	HTTPS /REST
Letter Manager API	Internal API for the Letter Manager	443	HTTPS /REST
ECM-L API	The API for the umbrella service for the ECM-L system, integrated with all of the internal components.	443	HTTPS /REST
FTI File Repo	FTI File Repository is the system of record for documents that contain Federal Tax Information and provides API access to those documents.	443	HTTPS /REST

Services Context (SvcV-1) (VASI APIs)

The following diagram describes service compositions and interactions, and any immediate dependencies. The diagram highlights external system services that connect to the focal architecture.

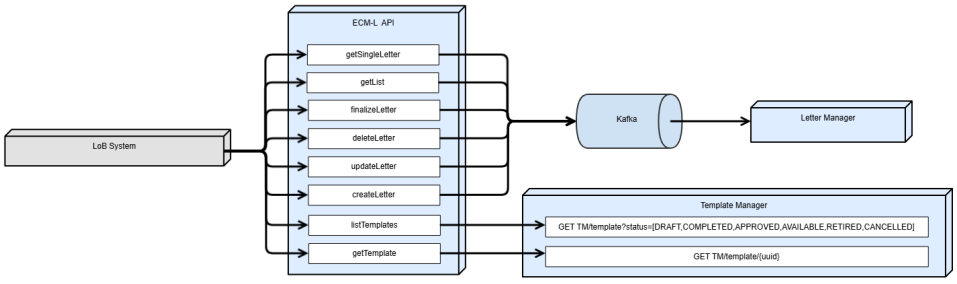


Legend
Gray Rectangle = External to core system(s) White = Internal to core system(s)

Service / API	Description
Resource Manager	Service providing LoB with the ability to create, test, replace, and retire Resources.
Template Manager	Service providing LoB with the ability to create, test and publish Changesets as well as create, test and retire Templates.
Letter Manager	Service providing ability to create Letter Instances, the Letter Manager will grab the Template and populate using Resources.
ECM-L API	ECM-L API provides the LoB system with access to the Letter Instances for sending and storing. ECM-L API is an outward-facing API for external applications to use to interact with the ECM-L components; it can Render Letter Instances as PDFs as well as handle the storage (in Claim Evidence) and distribution (via Package Manager) of generated PDFs.
LoB API	LoB API is a placeholder for the connection that will be set up for Resources to point to and access data from when populating a Template.

Services Resource Flow (SvcV-2)

The following diagram specifies the Resource Flows between Services and may also list the protocol stacks used in connections, providing a precise specification of a connection between Services.



Legend

Gray Rectangle = External to core system(s)

White = Internal to core system(s)

API Operations Lists

Below are lists of the available operations for the different API's in the ECM-L components.

Operational Diagrams

We have a growing library of diagrams mapping out individual ECM-L operations that you can view here: ECM-L Operational Diagrams.

Below are table-based representations of the operations for each component.

Resource Manager API Operations

Click arrow to View Resource Manager Operations List.

Template Manager API Operations

Click here to View Template Manager Operations List.

Letter Manager API Operations

Click here to View Letter Manager Operations.

ECM-L API Operations

Click here to View ECM-L API Operations.

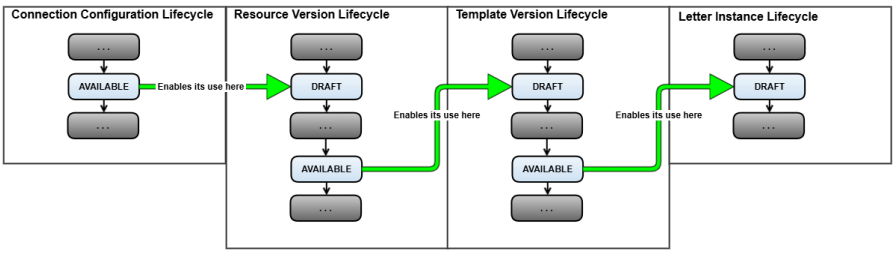
State Transitions (OV-6b)

The following diagrams map out the various states of being for individual ECM-L elements and the flow patterns between those states, as well any events that are triggered as these elements transition from one state to the next. These state transition diagrams represent the "heartbeat" of the ECM-L ecosystem and are grounded in an event-based architectural philosophy.

Lifecycle Interactions and Availability

This diagram helps to outline the concept of AVAILABILITY and how the different elements of the ECM-L components relate to one another in their Lifecycles.

Before an element of one of the ECM-L components is ready to be consumed and used, it must be made AVAILABLE.



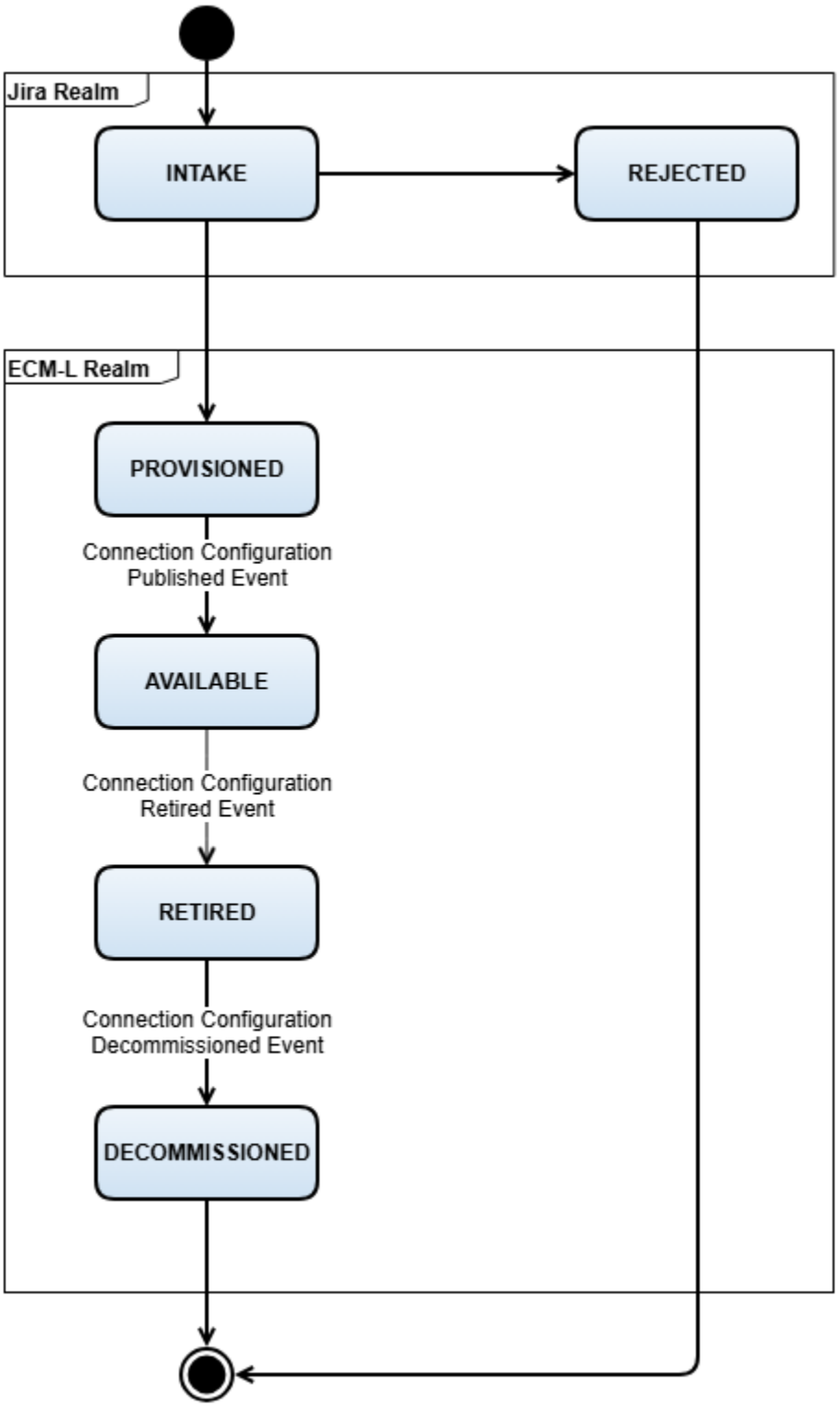
- Once a *Connection Configuration* is made AVAILABLE by a user, it can be used to create a new DRAFT *Resource Version*.
- Once a *Resource Version* is made AVAILABLE by a user, it can be used in the configuration of a new DRAFT *Template Version.*
- Once a *Template Version* is made AVAILABLE by a user, it can be used to create new DRAFT *Letter Instances*.

Resource Manager Elements

These are the lifecycle diagrams for elements belonging to the Resource Manager.

Connection Configuration Lifecycle

Connection configurations are entered into the system via an intake process through Jira, and then a user is able to move them through their lifecycle in the Resource Manager.



Jira Realm

These are not statuses that would surface within the ECM-L eco-system because they represent activity outside of it. Events do not take place here since these activities are not a part of the ECM-L components.

<table>
<tr><td>INTAKE</td><td>Dev Ops is processing the necessary intake documentation submitted by a Line of Business (LoB) to configure the ECM-L eco-system to interact with and integrate with the API submitted for the Connection Configuration.

Transitions from this state:
<table><tr><th>TO</th><th>DESCRIPTION</th></tr><tr><td>REJECTED</td><td>Something went wrong during the processing of the Intake Ticket and Line of Business (LoB) intervention is now needed to correct the information in the ticket.</td></tr><tr><td>PROVISIONED</td><td>The Dev Ops have completed what is necessary to validate the integration with the Connection Configuration's targeted API. The PROVISIONED Connection Configuration will be visible in the Resource Manager.</td></tr></table></td></tr>
<tr><td>REJECTED</td><td>The intake process was unsuccessful. The Line of Business and Dev Ops teams should work together to determine what caused the integration failure and work to address it.</td></tr>
</table>

ECM-L Realm

<table>
<tr><td>PROVISIONED</td><td>The Connection Configuration has made it through the intake process and is ready for a Line of Business Administrator to Test it and make it AVAILABLE for use. Should Testing fail, the LOB Admin will need to submit a Change Request for Dev Ops to fix the issue.

Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>ConnectionConfigurationPublishedEvent</td><td>AVAILABLE</td><td>The LoB Administrator has tested the PROVISIONED Connection and successfully connected from within the Resource Manager. The LOB Administrator has changed the Connection from PROVISIONED to AVAILABLE.</td></tr></table></td></tr>
<tr><td>AVAILABLE</td><td>The Connection Configuration has been properly PROVISIONED and Tested, and the LoB Administrator has made it AVAILABLE for use. While in this state the Connection Configuration is AVAILABLE to be used by Resources. We recommend Connection Configuration persist unchanged and be immutable once made AVAILABLE and has been used by Resources.

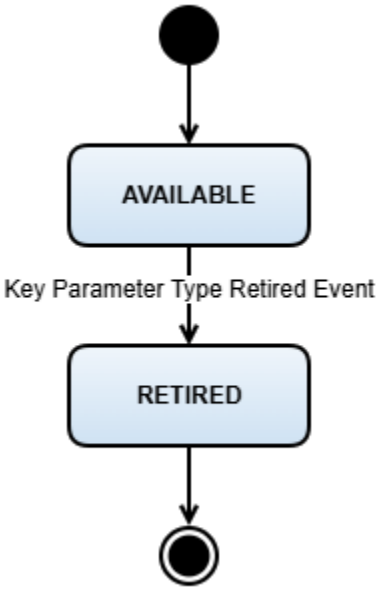
Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>ConnectionConfigurationRetiredEvent</td><td>RETIRED</td><td>An LoB Administrator has indicated that the Connection Configuration can no longer be used by the Resources configured to connect to it.</td></tr></table></td></tr>
<tr><td>RETIRED</td><td>No longer available for use by anything new.

The Connection Configuration Version can no longer be used in the configuration of new Resources.

Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>ConnectionConfigurationDecommissionedEvent</td><td>DECOMMISSIONED</td><td>An LoB Administrator will mark the Connection Configuration as Decommissioned. As a result of the decommissioning, the Connection will no longer respond/provide data to the Resources configured to connect to it and no new Resources can be configured to connect.</td></tr></table></td></tr>
<tr><td>DECOMMISSIONED</td><td>The API associated with this Connection Configuration is non-operational and can no longer respond to requests.</td></tr>
</table>

Key Parameter Type Lifecycle

Key Parameters have a very basic lifecycle where users in the Resource Manager create them to make them AVAILABLE, and then RETIRE them when they are no longer meant to be used.



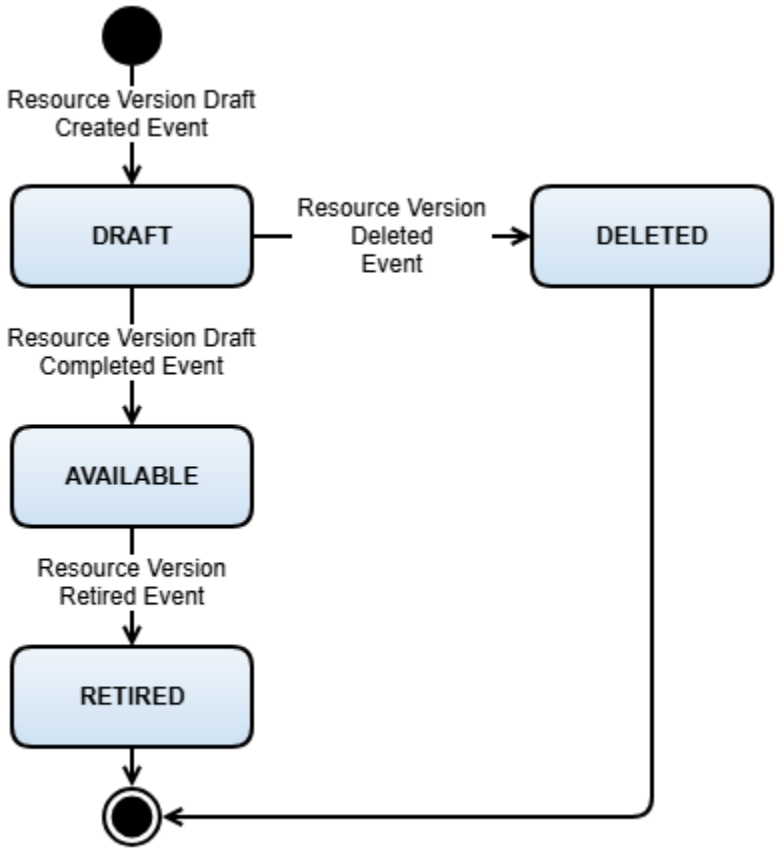
<table>
<tr><td>AVAILABLE</td><td>This Key Parameter Type has been added into the Resource Manager by a Line of Business (LoB) Administrator and is now AVAILABLE for use by Workflows and Resources.

Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>KeyParameterTypeRetired Event</td><td>RETIRED</td><td>Key Parameter Type is no longer needed and has been RETIRED by a LoB Administrator.</td></tr></table></td></tr>
<tr><td>RETIRED</td><td>No longer available for use by anything new.

This Key Parameter Type can no longer be selected as an option when choosing one during the configuration of a Workflow or a Resource.</td></tr>
</table>

Resource Version Lifecycle

Resource Versions are managed in the Resource Manager and can be worked on while in DRAFT, and then explicitly made AVAILABLE by a user when they are done. If a Resource Version is still in DRAFT it can be DELETED, and if we want to stop using one after it has been made AVAILABLE, we can move it to RETIRED.



<table>
<tr><td>DRAFT</td><td>This Resource Version has been created by a Line of Business (LoB) Administrator using an AVAILABLE Connection Configuration and is in the process of configuring it.

Transitions to this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>ResourceVersionDraftCreatedEvent</td><td>DRAFT</td><td>The Resource Version has been created in the DRAFT state by a LoB Administrator.</td></tr></table>
Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>ResourceVersionDeletedEvent</td><td>DELETED</td><td>The Resource Version was never made AVAILABLE for use, is no longer needed, and has been DELETED by an LoB Administrator.</td></tr><tr><td>ResourceVersionDraftCompletedEvent</td><td>AVAILABLE</td><td>The Resource Version has been completely configured and a LoB Administrator has made it AVAILABLE for use.</td></tr></table></td></tr>
<tr><td>AVAILABLE</td><td>This Resource Version is fully configured and a LoB Administrator has made it AVAILABLE for use in Template Versions.

Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>ResourceVersionRetiredEvent</td><td>RETIRED</td><td>The Resource Version is no longer needed, but has already been made AVAILABLE for use, so it is RETIRED.</td></tr></table></td></tr>
<tr><td>RETIRED</td><td>No longer available for use by anything new.

This Resource Version can no longer be used in the configuration of a new Template Version.</td></tr>
</table>

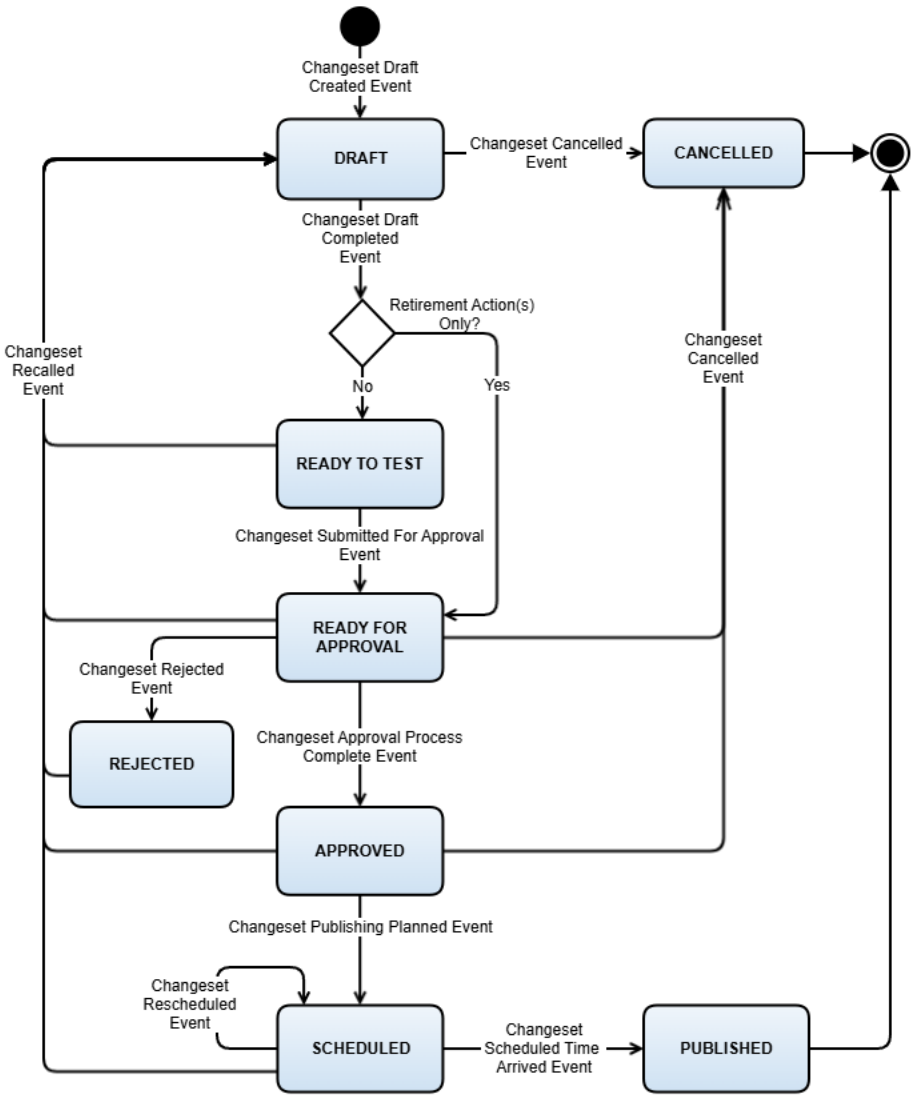
DELETED	This Resource Version was never made AVAILABLE for use, is no longer needed, and has been DELETED.

Template Manager Elements

These are the lifecycle diagrams for elements belonging to the Template Manager.

Changeset Lifecycle

Changesets are worked on while they are in a DRAFT state, and then can be moved to READY FOR TEST, where their content can be tested. Once all the testing is complete, they can be submitted for approval, moving them into the READY FOR APPROVAL state. After that they can be APPROVED, and then SCHEDULED for publication. Once the configured scheduling date arrives, the SCHEDULED Changeset is moved to PUBLISHED by the system, and its content becomes AVAILABLE for use.



DRAFT

This Changeset has been created, named, and work has begun to prepare it for publication. During this state, Templates are being created, added/removed to/from the Changeset, and tested. During this state the Changeset and its content remain modifiable. Test Scenarios can be created and executed to allow for incremental testing of a Template, but can not yet be marked as PASSED.

If a user has both the Changeset Editor and Changeset Publisher roles, they cannot mark their own Changeset as APPROVED.

Transitions to this state:

EVENT	TO	DESCRIPTION
ChangesetDraftCreatedEvent	DRAFT	The Changeset has been created in the DRAFT state by the Changeset Editor.

Transitions from this state:

EVENT	TO	DESCRIPTION
ChangesetDraftCompletedEvent	READY TO TEST	All Template Versions have been marked as Complete and the Changeset Editor has indicated that the Changeset is ready to be tested.
ChangesetCancellationProcessCompleteEvent	CANCELLED	The Changeset and all of its contents have been cancelled by the Changeset Editor and will no longer be moved towards publication.

READY TO TEST

This Changeset and all of the work within its contents has been completed, there are Template Versions within the Changeset that need to be marked as Passed, and the Changeset Editor has indicated that the Changeset is READY TO TEST. From this state, Template Versions must each have at least one Passed Test Scenario before the Changeset they contain can be submitted for approval.

This Status is skipped for a Changeset that contains only Retirement Actions, since Retirement Actions are not tested.

The Changeset Editor can continue to create and modify Test Scenarios, run them, and mark them as PASSED. The content of the Template Versions can not be modified unless the Changeset is Recalled back to the DRAFT state.

Transitions from this state:

EVENT	TO	DESCRIPTION
ChangesetSubmittedForApprovalEvent	READY FOR APPROVAL	All Test Scenarios for COMPLETED Template Version(s) have been created and have PASSED, and Changeset Editor has submitted the Changeset for Approval. Marking a Test Scenario as PASSED means the Changeset Editor has created a Test Scenario, executed it, viewed the results, and verified that the Rendered PDF looks as expected. The Changeset Editor can create multiple Test Scenarios with different data as needed, however all must be marked as PASSED before this transition can take place.
ChangesetRecalledEvent	DRAFT	This Changeset has been recalled to allow for additional modification.
ChangesetCancellationProcessCompleteEvent	CANCELLED	The Changeset and all of its contents have been cancelled by the Changeset Editor and will no longer be moved towards publication.

READY FOR APPROVAL

Every Template Version in the Changeset has at least one corresponding Test Scenario that has been marked as Passed, and a Changeset Editor has submitted the Changeset for Approval.

For Changesets that do not contain a Template Version, such as Changesets that only contain a Retirement Action, no Test Scenarios are required.

While within this state Test Scenarios can continue to be created, executed, and marked as PASSED to allow the Changeset Editor to create and run additional Test Scenarios.

Prior to Approving the Changeset, all of the contents of the Changeset, such as the Template Versions and Retirement Actions, must be individually APPROVED.

Transitions from this state:

EVENT	TO	DESCRIPTION
ChangesetApprovalProcessCompletedEvent	APPROVED	The Changeset and its contents has moved through the applicable Approval processes and has been APPROVED by a Changeset Publisher.
ChangesetRecalledEvent	DRAFT	This Changeset has been recalled by a Changeset Editor to allow for additional modification.
ChangesetRejectedEvent	REJECTED	Changeset Publisher reviewed and REJECTED the Changeset. Changeset Publisher must provide a Rejection Reason before the Changeset can transition to REJECTED state.
ChangesetCancellationProcessCompleteEvent	CANCELLED	The Changeset and all of its contents have been cancelled by a Changeset Publisher and will no longer be moved towards publication.

REJECTED

This Changeset Publisher has reviewed the contents of the Changeset and REJECTED it. They provide a Rejection Reason so that the Changeset Editor that owns the Changeset can review it and return it to the DRAFT state for additional modification.

Transitions from this state:

EVENT	TO	DESCRIPTION
ChangesetRecalledEvent	DRAFT	The Changeset Editor, after reviewing the REJECTED Changeset, moves it back to the DRAFT state for additional modification..

APPROVED

This Changeset Publisher has reviewed the contents of the Changeset and Approved it. All of the contents of the Changeset must be Approved before the Changeset itself can be APPROVED.

Transitions from this state:

EVENT	TO	DESCRIPTION
ChangesetRecalledEvent	DRAFT	Changeset Editor has determined that the APPROVED Changeset needs more work and so it is returned to the DRAFT state for additional modification.
ChangesetCancellationProcessCompleteEvent	CANCELLED	The Changeset and all of its contents have been cancelled by Changeset Publisher and will no longer be moved towards publication.
ChangesetPublishingPlannedEvent	SCHEDULED	The Changeset is set up to be Published, either immediately or for a specific date.

SCHEDULED

This Changeset Publisher has Scheduled it to be PUBLISHED at the Scheduled Date.

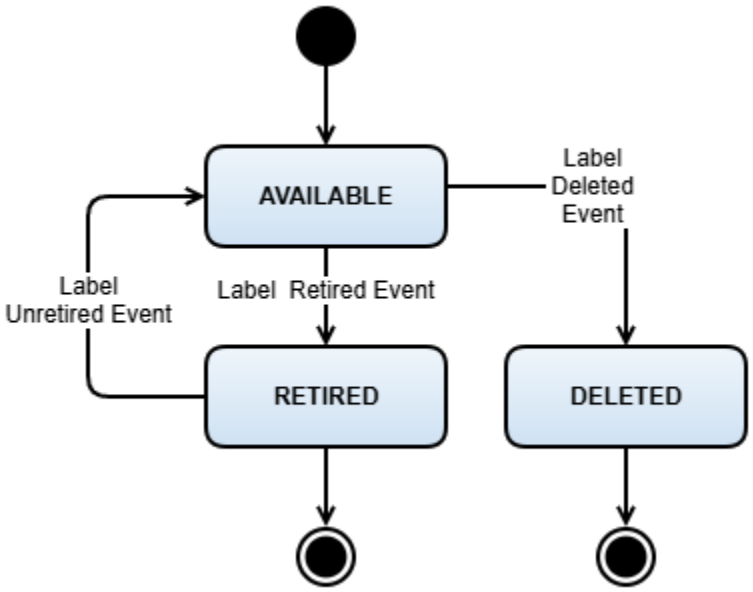
Transitions from this state:

EVENT	TO	DESCRIPTION
ChangesetRecalledEvent	DRAFT	Changeset Editor has determined that the SCHEDULED Changeset needs more work and so it is returned to the DRAFT state for additional modification.
ChangesetRescheduledEvent	SCHEDULED	Changeset Publisher has changed the Scheduled Date, which emits a Changeset Rescheduled Event.
ChangesetScheduledTimeArrivedEvent	PUBLISHED	The Scheduled time set for Publication has arrived and the Changeset is PUBLISHED and ready for use. This transition implies the transition of all new and included Templates to the AVAILABLE state and any AVAILABLE Templates targeted by a Retirement Action will transition to RETIRED.

PUBLISHED	This Changeset is now live and effective. All of the related Templates Versions are moved into the AVAILABLE and all of the related Retirement Actions are actualized by moving their respective Template Versions to the RETIRED status. AVAILABLE Templates are available to create Letter Instances.
cancelled	This Changeset was deemed unnecessary and has been cancelled. A PUBLISHED Changeset cannot be cancelled.

Label Lifecycle

Labels have a very basic lifecycle where users in the Template Manager create them to make them AVAILABLE, and then RETIRE them when they are no longer meant to be used.

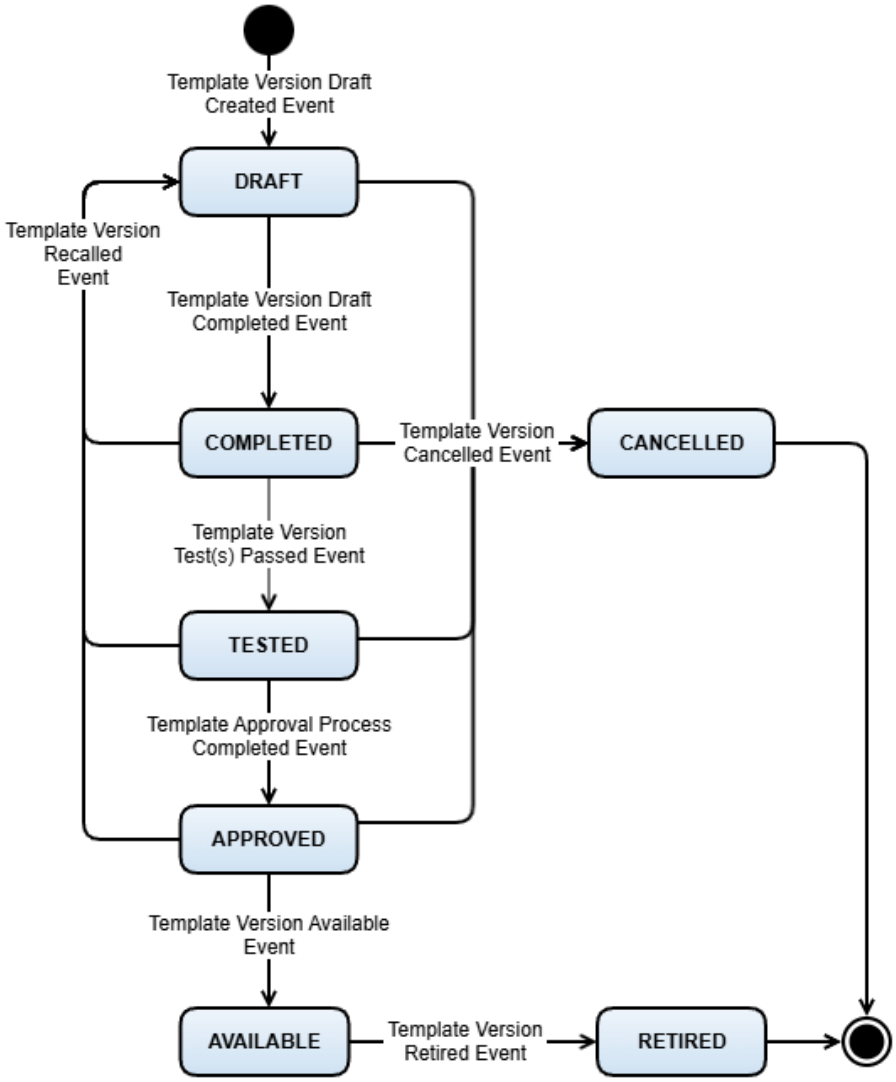


AVAILABLE	This Label has been added into the Template Manager and is now AVAILABLE for use. **Transitions from this state:** **EVENT** / **TO** / **DESCRIPTION** LabelRetiredEvent / RETIRED / Label is no longer needed and has been RETIRED. LabelDeletedEvent / DELETED / Label is not in use and can be completely discarded.
RETIRED	No longer available for use by anything new. This Label can no longer be selected as an option when choosing one. **Transitions from this state:** **EVENT** / **TO** / **DESCRIPTION** LabelUnretiredEvent / Available / Label is no longer RETIRED and has been made AVAILABLE again.
DELETED	The Label has been deleted, can only be done when the Label is not in use.

Template Version Lifecycle

Template Versions are moved through their lifecycle while within a Changeset. Users can create new DRAFT Template Versions and then, within a Changeset, mark them as COMPLETED. After the Changeset moves into the READY TO TEST, the user can create and run Test Scenarios to mark the Template Versions as TESTED. During the approval process, the person approving the Changeset is able to mark each individual Template Version as APPROVED. Once a Changeset is PUBLISHED, the Template Versions within it are moved to AVAILABLE.

If a Retirement Action for a Template Version is in a Changeset that is PUBLISHED, the Template Version moves into the RETIRED state.



DRAFT	This Template Version has been created, named, and is editable. Entering this state, an incremented Version number is assigned. Version numbers are unique to each individual Template Version. This is the only state in which the content and configuration of a Template Version can be modified. **Transitions to this state:** **EVENT** / **TO** / **DESCRIPTION** TemplateVersionDraftCreatedEvent / DRAFT / The Template Version has been created in the DRAFT state by the Changeset Editor. **Transitions from this state:** **EVENT** / **TO** / **DESCRIPTION** TemplateVersionDraftCompletedEvent / COMPLETED / The Changeset Editor is satisfied with their modifications to the Template Version and has marked it as COMPLETED. TemplateVersionCancelledEvent / CANCELLED / This Template Version is no longer needed and has been cancelled.

COMPLETED

This Template Version's updates and modifications are finished and the Changeset Editor has marked it as COMPLETED. The content and the configuration of the Template Version can no longer be modified.

Transitions from this state:

EVENT	TO	DESCRIPTION
TemplateVersionRecalledEvent	DRAFT	The Changeset Editor has decided that this Template Version needs more work and moves it back to DRAFT for further modification.
TemplateVersionTestPassedEvent	TESTED	All of the Test Scenarios for this Template Version have been run and have been marked as PASSED.
TemplateVersionCancelledEvent	CANCELLED	This Template Version is no longer needed and has been cancelled.

TESTED

This Template Version's Test Scenarios have all been reviewed and marked as PASSED by a Changeset Editor.

New Test Scenarios can be added to this Template Version while the Changeset is in the Ready for Approval state if the Changeset Publisher decides to add in additional testing during the approval process.

In this state the content and configuration of the Template Version can no longer be modified. If a user wishes to make additional modifications to the Template Version they must Recall it back to the DRAFT state.

The Changeset Publisher must first reject the Changeset, and then the Changeset Editor can recall the Changeset to the DRAFT state, which allows for the Template Version to be recalled back to DRAFT and cancelled.

Transitions from this state:

EVENT	TO	DESCRIPTION
TemplateVersionApprovalProcessCompletedEvent	APPROVED	The Template Version has gone through the applicable approval process and been APPROVED.
TemplateVersionRecalledEvent	DRAFT	The Changeset Editor has decided that this Template Version needs more work and moves it back to DRAFT for further modification.
TemplateVersionCancelledEvent	CANCELLED	This Template Version is no longer needed and has been cancelled.

APPROVED

This Template Version has passed testing and been approved by the Changeset Publisher. Test Scenarios can no longer be added to this Template Version.

In this state it can no longer be modified. If a Changeset Editor wishes to make additional modifications to it they must Recall it back to the DRAFT state.

The Changeset Publisher must first reject the Changeset, and then the Changeset Editor can recall the Changeset to the DRAFT state, which allows for the Template Version to be recalled back to DRAFT and cancelled.

Transitions from this state:

EVENT	TO	DESCRIPTION
TemplateVersionAvailableEvent	AVAILABLE	This Template Version was in a Changeset that has been PUBLISHED.
TemplateVersionCancelledEvent	CANCELLED	This Template Version is no longer needed and has been cancelled.
TemplateVersionRecalledEvent	DRAFT	The Changeset Editor has reviewed Template Version and recalls the Changeset for modifications. The Changeset Editor has determined that the Template Version needs more work and moves it back to DRAFT for further modification.

<table>
<tr><td>AVAILABLE</td><td>This Template Version has been Published and is live and AVAILABLE for the instantiation of Letter Instances.

In this state it can no longer be modified. It can also no longer be Recalled.

If a user wishes to make additional modifications to it they must create a new Version.

⚠ Only the most recently Published Template Version will be flagged in Template Manager to indicate that it is the Template Version AVAILABLE in the Letter Manager to create Letter Instances.

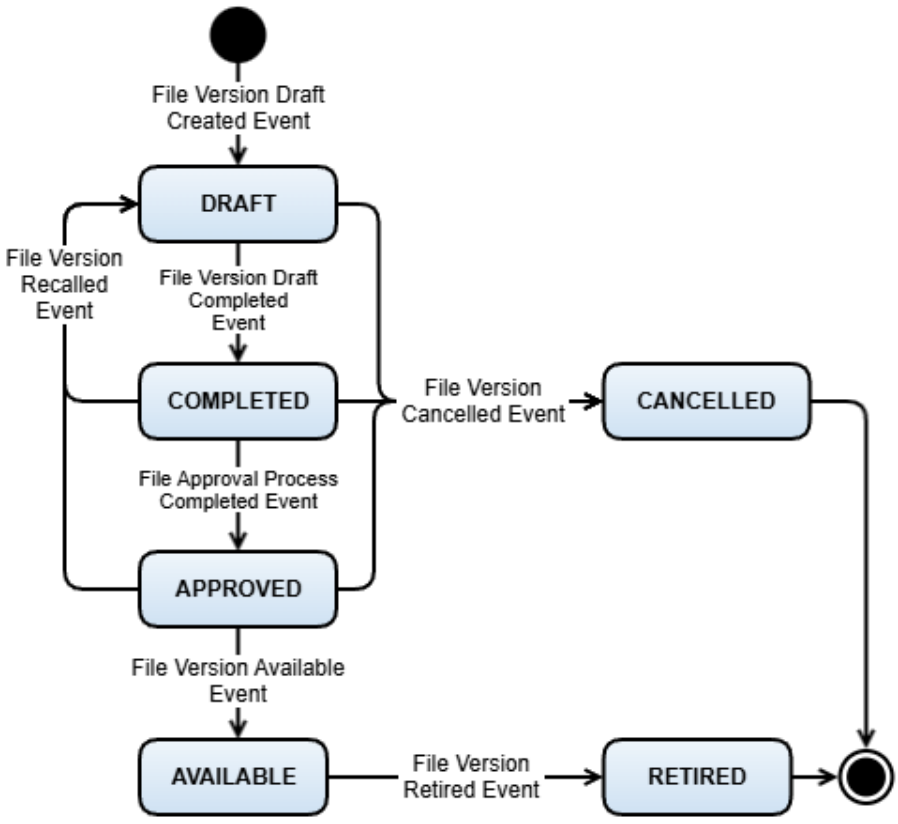
Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>TemplateVersion RetiredEvent</td><td>RETIRED</td><td>A Retirement Action targeting this Template Version has been published through a Changeset that results in this Template Version moving into a RETIRED state.</td></tr></table></td></tr>
<tr><td>CANCELLED</td><td>This Template Version is no longer needed and has been cancelled.</td></tr>
<tr><td>RETIRED</td><td>No longer available for use by anything new.

This Template Version can no longer be used to instantiate new Letter Instances.</td></tr>
</table>

File Version Lifecycle

File Versions are moved through their lifecycle while within a Changeset. Users can create new DRAFT File Versions and then, within a Changeset, mark them as COMPLETED. File Versions, unlike Template Versions, do not require formal testing. During the approval process, the person approving the Changeset is able to mark each individual File Version as APPROVED. Once a Changeset is PUBLISHED, the File Versions within it are moved to AVAILABLE.

If a Retirement Action for a File Version is in a Changeset that is PUBLISHED, the File Version moves into the RETIRED state.



DRAFT

This File Version has been created, named, and is editable. Entering this state, an incremented Version number is assigned. Version numbers are unique to each individual File Version. This is the only state in which the content and configuration of a File Version can be modified.

Transitions to this state:

EVENT	TO	DESCRIPTION
FileVersionDraftCreatedEvent	DRAFT	The File Version has been created in the DRAFT state by the Changeset Editor.

Transitions from this state:

EVENT	TO	DESCRIPTION
FileVersionDraftCompletedEvent	COMPLETED	The Changeset Editor is satisfied with their modifications to the File Version and has marked it as COMPLETED.
FileVersionCancelledEvent	CANCELLED	This File Version is no longer needed and has been cancelled.

COMPLETED

This File Version's updates and modifications are finished and the Changeset Editor has marked it as COMPLETED.

In this state it can no longer be modified. If a Changeset Editor wishes to make additional modifications to it they must Recall it back to the DRAFT state.

Transitions from this state:

EVENT	TO	DESCRIPTION
FileVersionRecalledEvent	DRAFT	The Changeset Editor has decided that this File Version needs more work and moves it back to DRAFT for further modification.
FileVersionCancelledEvent	CANCELLED	This File Version is no longer needed and has been cancelled.
FileVersionApprovedEvent	APPROVED	This File Version has been COMPLETED and can moved to APPROVED.

APPROVED

This File Version has been APPROVED by the Changeset Publisher.

In this state it can no longer be modified. If additional modifications are needed the File Version must be Recall back to a DRAFT state.

Unlike Template Versions, there is no need for testing a File Version.

Transitions from this state:

EVENT	TO	DESCRIPTION
FileVersionAvailableEvent	AVAILABLE	This File Version was in a Changeset that has been PUBLISHED.
FileVersionRecalledEvent	DRAFT	The Changeset Editor has decided that this File Version needs more work and moves it back to DRAFT for further modification.
FileVersionCancelledEvent	CANCELLED	This File Version is no longer needed and has been cancelled.

<table>
<tr><td>AVAILABLE</td><td>This File Version has been PUBLISHED via Changeset and is available for use.

In this state it can no longer be modified. It can also no longer be Recalled.

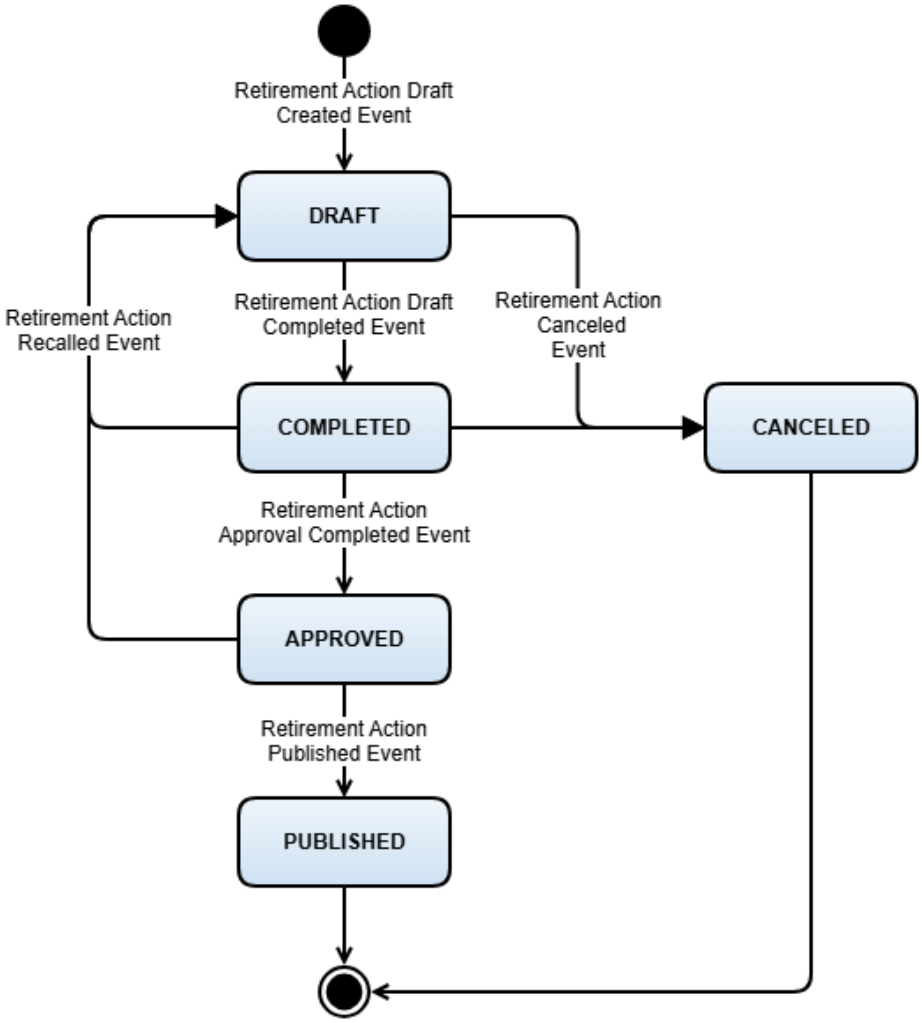
If a Changeset Editor wishes to make additional modifications to it they must create a new Version.

Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>FileVersionRetiredEvent</td><td>RETIRED</td><td>A Retirement Action targeting this File Version has been published through a Changeset that results in this File Version moving into a RETIRED state.</td></tr></table></td></tr>
<tr><td>cancelled</td><td>This File Version is no longer needed and has been cancelled.</td></tr>
<tr><td>RETIRED</td><td>No longer available for use by anything new.

This File Version can no longer be used in new Template Versions.</td></tr>
</table>

Retirement Action Lifecycle

Retirement Actions are created in the DRAFT state by the Changeset Editor to retire a Template or File Version. Once completed, they are submitted for approval; if approved they move to PUBLISHED and retire their contents, otherwise they can be rejected, recalled for changes, or cancelled before publication. A Changeset can contain multiple Retirement Actions.



<table>
<tr><td>DRAFT</td><td>This Retirement Action is created by a Changeset Editor and added to a Changeset. During this state the Changeset Editor is adding an AVAILABLE Template Version or File Version to retire.

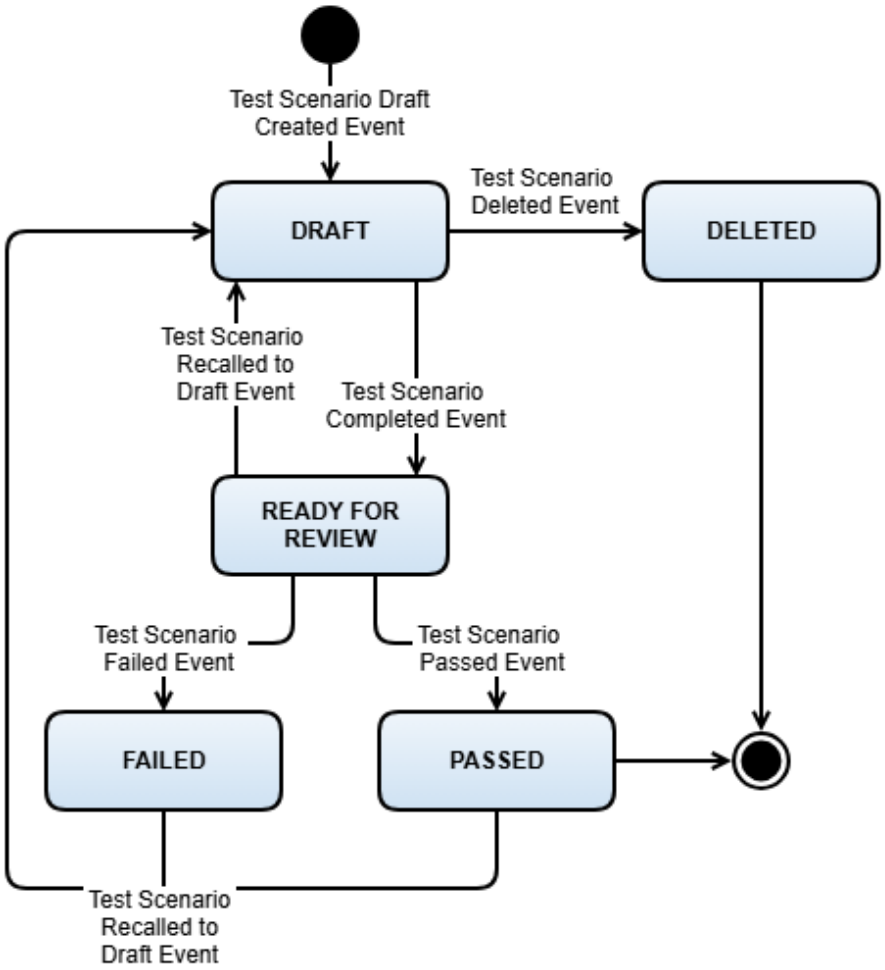
Transitions to this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>RetirementActionDraftCreatedEvent</td><td>DRAFT</td><td>The Retirement Action has been created in the DRAFT state by the Changeset Editor.</td></tr></table>
Transitions from this state:
<table><tr><td>RetirementActioncancelledEvent</td><td>cancelled</td><td>The Retirement Action content has been cancelled and will no longer be moved towards publication.</td></tr><tr><td>RetirementActionDraftCompletedEvent</td><td>COMPLETED</td><td>The Retirement Action has been created and is ready to be APPROVED.</td></tr></table></td></tr>
<tr><td>COMPLETED</td><td>In this state the Changeset Editor has added the item they decided to RETIRE, they mark the Retirement Action as COMPLETED, indicating the item is ready to be APPROVED by the Changeset Publisher.

Transitions from this state:
<table><tr><td>RetirementActionApprovalCompletedEvent</td><td>APPROVED</td><td>Retirement Action is reviewed for Retirement, the Changeset Publisher agrees with the action, so it is APPROVED.</td></tr><tr><td>RetirementActionRecalledEvent</td><td>DRAFT</td><td>The Changeset Editor has decided to change the Retirement Reason of the Retirement Action and moves it back to DRAFT for further modification.</td></tr><tr><td>RetirementActioncancelledEvent</td><td>cancelled</td><td>The Retirement Action content has been cancelled and will no longer be moved towards publication.</td></tr></table></td></tr>
<tr><td>APPROVED</td><td>Retirement Action has been reviewed for Retirement and APPROVED by Changeset Publisher.

Transitions from this state:
<table><tr><td>RetirementActionPublishedEvent</td><td>PUBLISHED</td><td>The Scheduled time set for Publication has arrived and the Changeset is PUBLISHED. This transition also moves the Retirement Action to PUBLISHED and the content transitions from AVAILABLE to a RETIRED state.</td></tr><tr><td>RetirementActionRecalledEvent</td><td>DRAFT</td><td>The Changeset Editor has decided that the content of the Retirement Action needs more work and moves it back to DRAFT for further modification.</td></tr></table></td></tr>
<tr><td>PUBLISHED</td><td>The Retirement Action is actualized and the Templates Version or File Version is moved to a RETIRED state.</td></tr>
<tr><td>cancelled</td><td>The Retirement Action has been cancelled and will no longer be moved towards publication.</td></tr>
</table>

Test Scenario Lifecycle

Test Scenarios are unique to Template Versions and can be worked on while they are in the DRAFT state. Once a user is finished working on a Test Scenario they can mark it as READY FOR REVIEW. It can then be marked as PASSED or FAILED, or recalled back to DRAFT from these states.



DRAFT

A Changeset Editor has created a Test Scenario for a Template Version and is in the process of configuring it, filling out all of the fields necessary to hydrate the document for rendering it based on the provided input. While in this state, the Changeset Editor can create and run Test Scenarios for Template Versions and can view the rendered PDF. However, Test Scenarios cannot be marked as PASSED or FAILED in the DRAFT state. The option to mark the Scenario as PASSED or FAILED is only available once the Scenario moves to the 'Ready for Review' state.

Transitions to this state:

EVENT	TO	DESCRIPTION
TestScenarioDraftCreatedEvent	DRAFT	The Test Scenario has been created in the DRAFT state by the Changeset Editor.

Transitions from this state:

EVENT	TO	DESCRIPTION
TestScenarioDeletedEvent	DELETED	Test Scenario is deemed no longer needed and is DELETED by a Changeset Editor.
TestScenarioCompletedEvent	READY FOR REVIEW	Test Scenario has been successfully run and the results are READY FOR REVIEW.

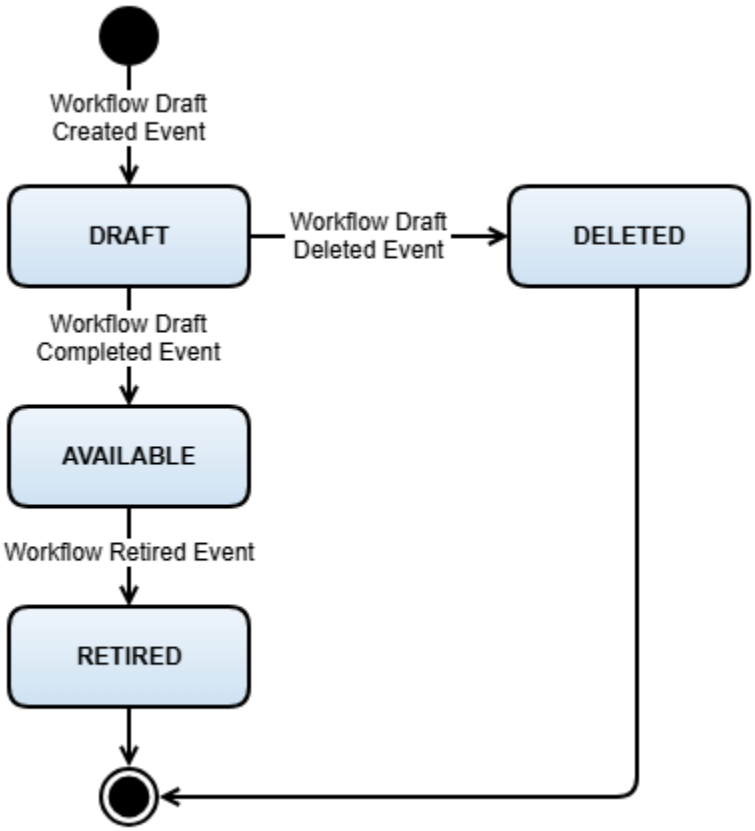
<table>
<tr><td>READY FOR REVIEW</td><td>In the READY FOR REVIEW state, the Changeset Editor evaluates the rendered PDF results produced by the Test Scenario execution and marks the Test Scenario as PASSED or FAILED, they may recall the Test Scenario for further editing.

Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>TestScenarioFailedEvent</td><td>FAILED</td><td>The Changeset Editor has reviewed the rendered PDF results and has marked the Test Scenario as FAILED.</td></tr><tr><td>TestScenarioPassedEvent</td><td>PASSED</td><td>The Changeset Editor has reviewed the rendered PDF results and has marked the Test Scenario as PASSED.</td></tr><tr><td>TestScenarioRecalledtoDraftEvent</td><td>DRAFT</td><td>Test Scenario has been recalled for additional modification or review.</td></tr></table></td></tr>
<tr><td>FAILED</td><td>After evaluation by a Changeset Editor, the Test Scenario has been deemed unacceptable. From here, the Test Scenario can be recalled back to DRAFT, by either the Changeset Editor, for further modification.

Transitions from this state:
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>TestScenarioRecalledEvent</td><td>DRAFT</td><td>The Changeset Editor recalled the Failed Test Scenario back to DRAFT to make changes as needed.</td></tr></table></td></tr>
<tr><td>PASSED</td><td>After evaluation by a Changeset Editor, the Test Scenario has been deemed acceptable and has been marked as PASSED. If necessary, the Test Scenario can also be recalled back to DRAFT, by either the Changeset Editor, for further modification.
<table><tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr><tr><td>TestScenarioRecalledEvent</td><td>DRAFT</td><td>The Changeset Editor recalled the Passed Test Scenario back to DRAFT to make changes as needed.</td></tr></table></td></tr>
<tr><td>DELETED</td><td>This Test Scenario is no longer needed and has been DELETED.</td></tr>
</table>

Workflow Lifecycle

The Workflow Lifecycle is pretty basic and managed via the Template Manager. Users are able to either create a DRAFT, mark it as AVAILABLE, or RETIRE them when they are no longer meant to be used. If a Workflow is still in the DRAFT state it can be DELETED.



<table>
<tr>
<td>DRAFT</td>
<td>The Workflow has been created in a DRAFT state by a Line of Business (LoB) Administrator. In this state, Name and Key Parameter are required to create a Workflow. Additional configurations, such as adding Full Templates and Storage and Distribution options, can optionally be included.

Transitions to this state:

<table>
<tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr>
<tr><td>WorkflowDraftCreatedEvent</td><td>DRAFT</td><td>The Workflow has been created in the DRAFT state by the LoB Administrator.</td></tr>
</table>
Transitions from this state:

<table>
<tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr>
<tr><td>WorkflowDraftCompletedEvent</td><td>AVAILABLE</td><td>The Workflow has been created and marked by the LoB Administrator as available for use.</td></tr>
<tr><td>WorkflowDraftDeletedEvent</td><td>DELETED</td><td>The Workflow has been deleted by the LoB Administrator and is no longer needed.</td></tr>
</table>
</td>
</tr>
<tr>
<td>AVAILABLE</td>
<td>This Workflow has been created by a Line of Business (LoB) Administrator and is AVAILABLE for use. In this state Templates can be aligned to the Workflow and Letter Instances can be created within it.

Transitions from this state:

<table>
<tr><th>EVENT</th><th>TO</th><th>DESCRIPTION</th></tr>
<tr><td>WorkflowRetiredEvent</td><td>RETIRED</td><td>Workflow is no longer needed and RETIRED from use.</td></tr>
</table>
</td>
</tr>
<tr>
<td>RETIRED</td>
<td>No longer available for use by anything new.

No new Letter Instances can be created within this Workflow, and no more Templates can be aligned to it.</td>
</tr>
</table>

DELETED	The Workflow has been deleted by the LoB Administrator is no longer available.

Letter Manager Elements

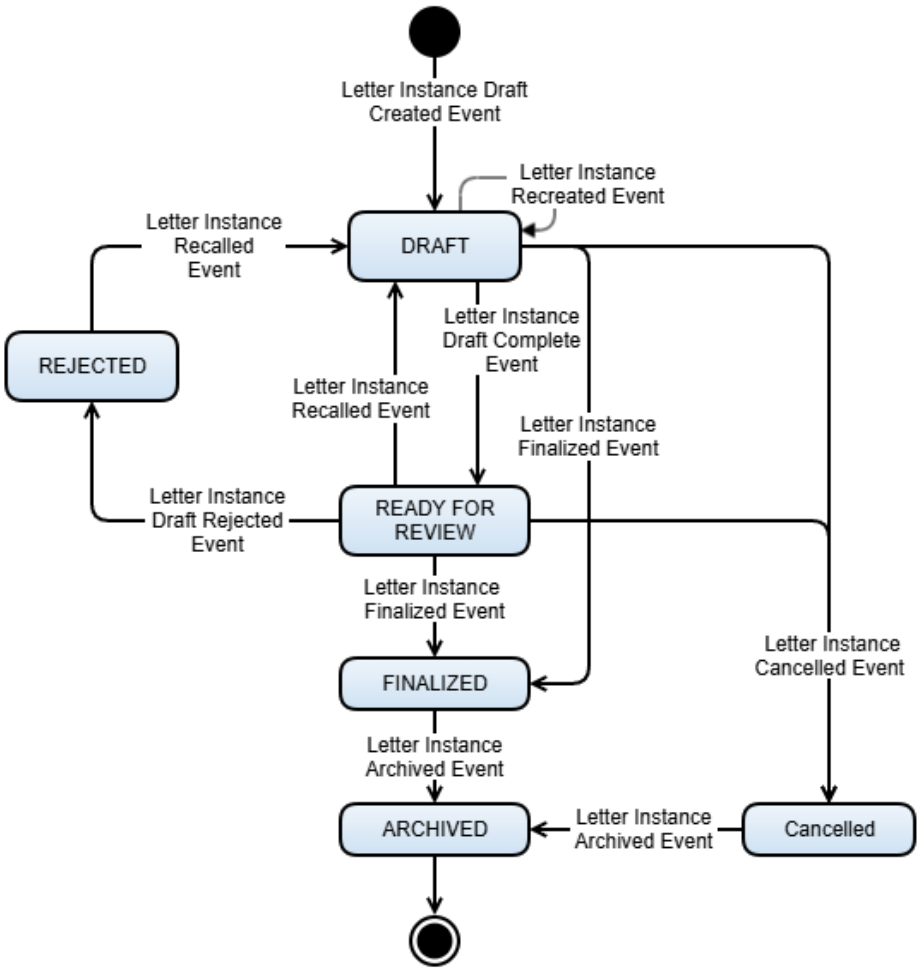
These are the lifecycle diagrams for elements belonging to the Letter Manager.

Letter Instance Lifecycle

Letter Instances begin in the DRAFT state, and will remain there while they are being modified by a user. When a user is happy with their modifications they can submit it for READY FOR REVIEW, and then after review and approval, it can be moved to FINALIZED.

With the right permissions, users are able to bypass the READY FOR REVIEW state, to accommodate automated and system users that need to generate correspondence without human intervention, as well as Lines of Business that do not have an approval process for Letter Instances.

As a part of our archival process, the Letter Instance can move into the ARCHIVED state. (See Letter Instance Archival.)



DRAFT

This Letter Instance has been instantiated based on an AVAILABLE Template Version. Upon entering and exiting from this state, the ECM-L system will pull in everything needed from Resources configured in the Template Version. In the Letter Manager, when a Letter Editor is creating a new Letter Instance, it will always be instantiated based on the most recent AVAILABLE Template Version. System Actors are able to generate Letter Instances based on older AVAILABLE Template Versions so that they have access to a reliable API data model.

When a user has both the Letter Editor and Letter Finalizer roles assigned, they are able to move a DRAFT Letter Instance directly to the FINALIZED state, effectively bypassing the READY FOR REVIEW state.

DRAFT Letter Instances can be rendered as a PDF but will be contain a "DRAFT" watermark.

Transitions to this state:

EVENT	TO	DESCRIPTION
LetterInstanceDraftCreatedEvent	DRAFT	The Letter Instance has been created in the DRAFT state by the Letter Editor.

Transitions from this state:

EVENT	TO	DESCRIPTION
LetterInstanceRecreatedEvent	DRAFT	This Letter Instance has been re-instantiated based on a newer Version of the Template. This is optional after a Changeset Publisher has published a newer Version of the Template. If a newer Template Version becomes AVAILABLE while the Letter Instance is in this state (i.e., Changeset containing Template Version X+1 published while Letter Instance based on Template Version X in DRAFT state), Letter Editor is given the option to Recreate the Letter Instance using the newer Version of the Template. If the Letter Editor chooses to take this action, any DRAFT content will be deleted.
LetterInstanceCancelledEvent	CANCELLED	Letter Instance no longer needed and has been CANCELLED.
LetterInstanceDraftCompleteEvent	READY FOR REVIEW	Letter Instance based on AVAILABLE Template Version created and marked READY FOR REVIEW.
LetterInstanceFinalizedEvent	FINALIZED	The Letter Instance has been reviewed, and the resulting rendered PDF has been deemed acceptable. U*ser (s) that have both the Letter Editor and Letter Finalizer roles are able to "Fast Track" directly from DRAFT to AVAILABLE state.*

READY FOR REVIEW

This Letter Instance contains all elements required by the Template Version (e.g., inputs, attachments, inclusions, etc.) and has been submitted for review.

> For some Lines of Business a single user will move the Letter Instance all the way through the lifecycle to Finalization without a separate user approving the Letter Instance.

> READY FOR REVIEW Letter Instances can be rendered as a PDF but will contain a "READY FOR REVIEW" watermark.

Transitions from this state:

EVENT	TO	DESCRIPTION
LetterInstanceDraftRejectedEvent	REJECTED	Changes need to be made to the Letter Instance before it can pass review and start the finalization process, so a Letter Finalizer has REJECTED the Letter Instance to DRAFT for further modification.
LetterInstanceRecalledEvent	DRAFT	This Letter Editor has decided that the Letter Instance needs further modifications and moves it back to the DRAFT state to pull out of the review process.
LetterInstanceFinalizedEvent	FINALIZED	The Letter Instance has been reviewed, and the resulting rendered PDF has been deemed acceptable.
LetterInstanceCancelledEvent	CANCELLED	Letter Instance no longer needed and has been CANCELLED

REJECTED

This Letter Instance has been REJECTED for further modification. In this state the Letter Instance will be recalled back to the DRAFT state.

Transitions from this state:

EVENT	TO	DESCRIPTION
LetterInstanceRecalledEvent	DRAFT	The Letter Finalizer has reviewed the Letter Instance and it has been REJECTED for further modifications. The Letter Instance will need to be recalled to DRAFT state by the Letter Editor for modifications.

FINALIZED

This Letter Instance has been approved and made Final. In this state the Letter Instance is immutable and will persist unchanged for historical purposes, and every subsequent rendering of the Letter Instance will have the same result for all time.

> FINALIZED Letter Instances can be rendered as a PDF with no watermark.

CANCELLED

This Letter Instance is no longer needed and has been CANCELLED.

> CANCELLED Letter Instances can no longer be rendered.

ARCHIVED

After a given period of time, the Letter Instance will be ARCHIVED as per applicable archival procedure.

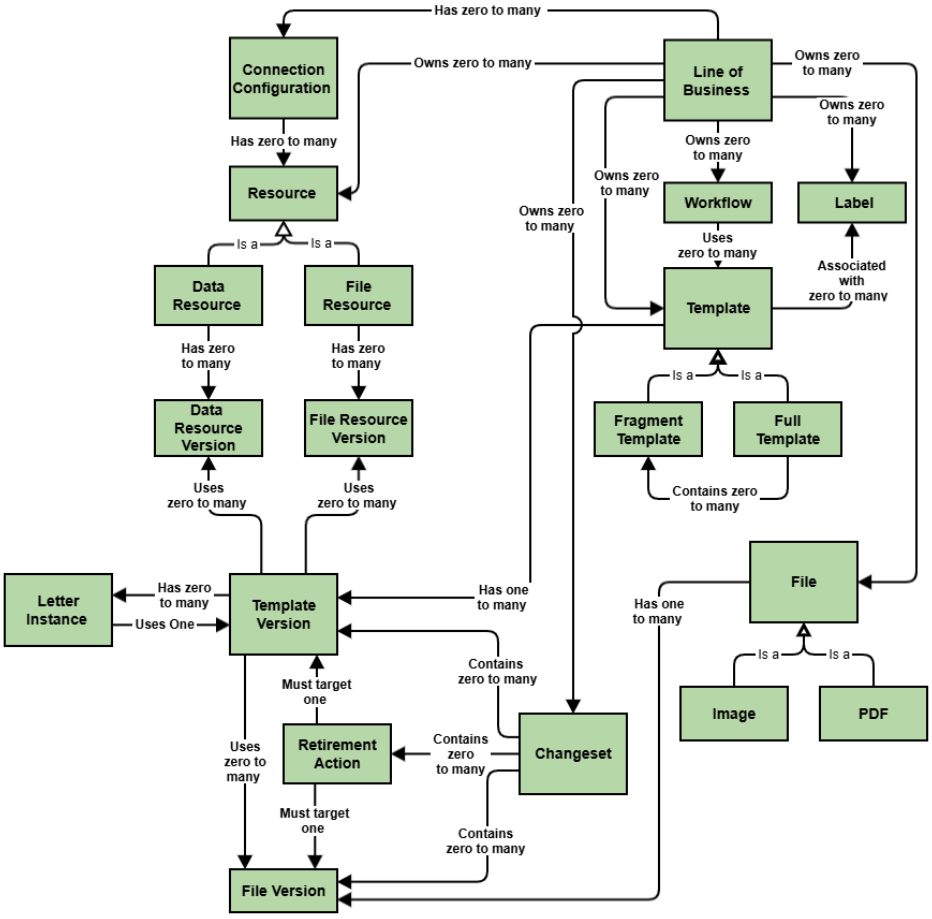
Data Models

Conceptual Data Models (DIV-1)

The following diagrams capture high level data concepts of ECM-L.

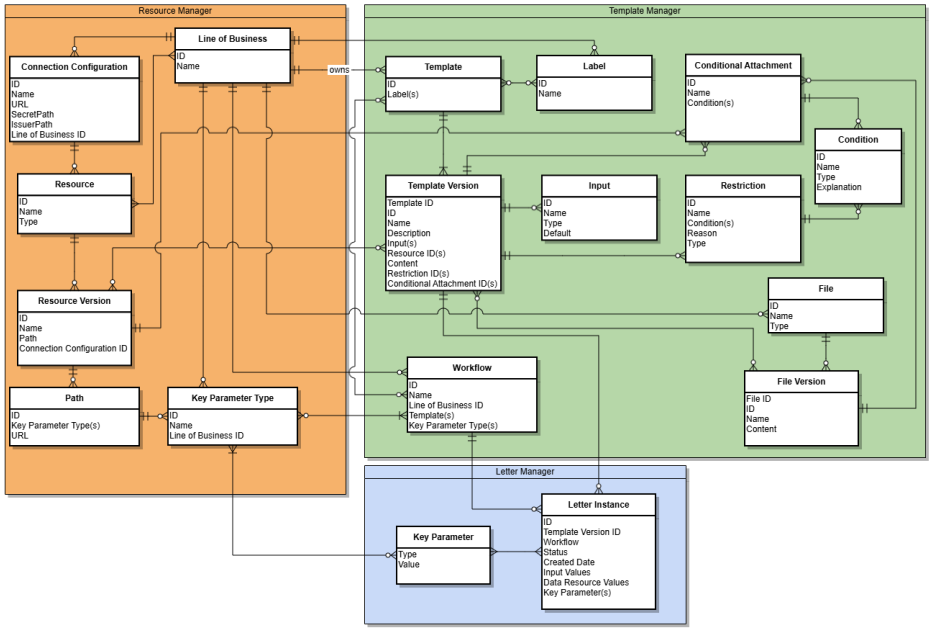
Key Concepts

This is a diagram illustrating the basic high level concepts and how they relate to one another.



Entity Relationship Diagram

This Entity Relationship Diagram illustrates how the high level concepts of the ECM-L systems interact with and relate to one another.



Resource Manager

Line of Business

An organization serving a specific customer need or business function. Changesets, Template Versions, Workflows, Resources, Key Parameter Types, and Connection Configurations all belong to a specific Line of Business.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
CSS Identifier	An abbreviated name of the Line of Business (6 character max) for CSS Functions.

Connection Configuration

All of the configuration information necessary for the ECM-L to successfully interact and integrate with a specific external API. These are loaded into the Resource Manager via a configuration file and then are made available for use by Resources via the Resource Manager.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
URL	The environment-specific URL for the API.
Secret Path	The path to the secret in Vault.
Issuer Path	The path to the issuer in Vault.
Line of Business ID	The UUID of the Line of Business to which it belongs.

Resource

The information necessary to gather content and data from outside of the ECM-L system that can be used within the documents that it creates and renders.

Attribute Name	Description
ID	Its immutable UUID.
Name	The user friendly name of a Resource.

Type	The type of the Resource.

Resource Version

A specific Version of a Resource.

Attribute Name	Description
ID	Its immutable UUID.
Paths	A list of paths that can be used to access the Resource, depending on the given Key Parameter Type.
Connection Configuration ID	The ID of the Connection Configuration it uses.

Path

The information related to a specific operation / API path that will allow the ECM-L System to access the information in a Resource Version.

Attribute Name	Description
ID	Its immutable UUID.
Key Parameter Type(s)	The IDs of the Key Parameters required for this path.
URL	The path to the endpoint in the API represented by the chosen Connection Configuration.

Key Parameter Type

A Line of Business specific data type used to organize and contextualize correspondence and used as a Parameter to gather contextual data from a Resource.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
Line of Business ID	The UUID of the Line of Business to which it belongs.

Template Manager

Workflow

The Workflow or sub process in which Letters are created within a given Line of Business.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
Line of Business ID	The UUID of the Line of Business to which it belongs.
Templates	The UUIDs of any Templates associated with this Workflow.
Key Parameter Type(s)	The type(s) of Key Parameter given for the context.

Template Label

A label is used to help users assign meaning and organizational order to their Templates.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.

File

A piece of file-based content. These are versioned and each File Version is managed by a Changeset, similar to Templates. There are two different types of Files - PDF's and Images

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
Type	The type of the File - either a PDF or an Image.

File Version

The versioned File content.

Attribute Name	Description
File ID	The immutable UUID of the Filethis is Versioning.
ID	Its immutable UUID.
Name	Its user-friendly name.
Content	The actual content of the File.

Template

A collection of versioned configuration information and content required to manage a specific type of correspondence. There are two different types of Templates - Full Templates and Fragment Templates.

Attribute Name	Description
ID	Its immutable UUID.
Labels	The labels to which this Template has been aligned.

Template Version

The versioned configuration information and content required to manage a specific type of correspondence.

Attribute Name	Description
Template ID	The immutable UUID of the Template this is Versioning.
ID	Its immutable UUID.
Name	Its user-friendly name.
Inputs	A collection of inputs that can be used to hydrate the content of the Template with user created data.
Resource IDs	The UUID's of any Resources that are aligned to this Template, that can be used to hydrate the content of the Template with contextual data.
Content	The actual FreeMarker and textual content of the Template.
Restriction IDs	The UUID's of any restrictions that should be applied to correspondence created based on this Template.
Conditional Attachment IDs	The UUID's of any attachments that should be conditionally added to correspondence created based on this Template.

Conditional Attachment

Files like PDF's that are attached to correspondence if certain conditions are met.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
Conditions	A list of conditions that once met will result in it being attached to the correspondence.

Condition

A comparison or check used in Restrictions and Conditional Attachments.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
Type	The condition type, such as a string-match or a boolean condition.
Explanation	A user friendly explanation that describes the condition in plain English.

Restriction

A condition based restriction on creating correspondence related to a specific Template Version.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
Conditions	A list of the UUIDs for conditions that if met result in a Template Version being restricted.
Reason	A user-friendly description of the reason for this restriction.
Type	Determines the impact of this restriction - such as hiding it from the list of available Templates, or disabling it and indicating why using the given Reason.

Input

A way to collect data from a user that can be used to hydrate the content of a Template.

Attribute Name	Description
ID	Its immutable UUID.
Name	Its user-friendly name.
Type	Determines the type of input, which informs things like the formatting options in the Content Editor of the Template Manager and which form elements to present to the user in the Enterprise Letter Virtual Interview Screen (ELVIS).
Default	The optional default value of the Input.

Letter Manager

Letter Instance

An instantiated Letter based on the content and configuration options outlined in a specific version of a Full Template.

Attribute Name	Description
ID	Its immutable UUID.
Template Version ID	The UUID of the Template Version for which this instance was created and will be rendered against.
Workflow	The Workflow in which this Letter Instance was created.
Created Date	The date this instance was created.
Input Values	A collection of the data collected from users based on the Inputs defined in the Template Version.
Data Resource Values	A collection of the data collected from data resources based on the Resources defined in the Template Version.

Key Parameter

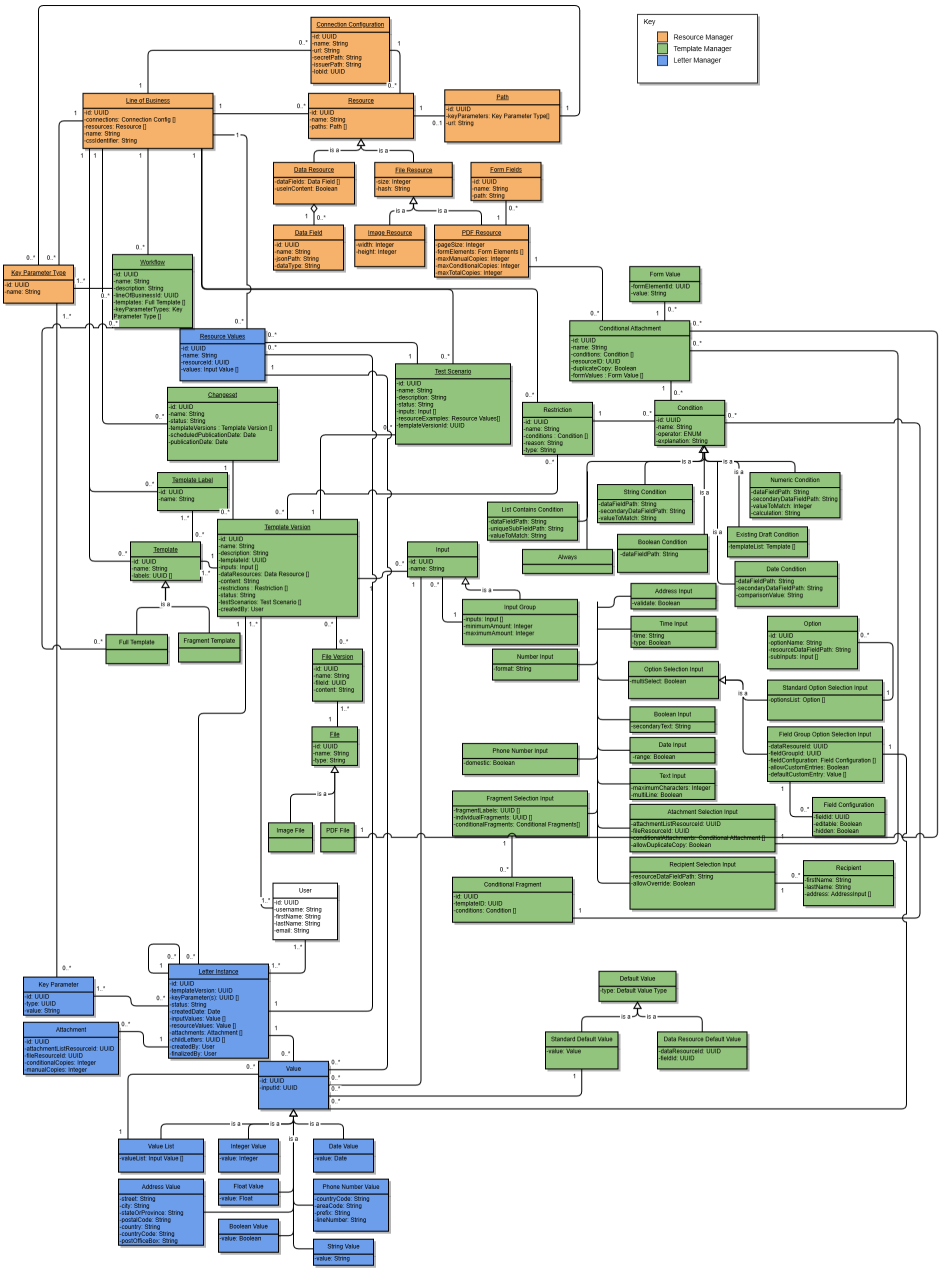
A type / value pair that informs us of the context in which a Letter Instance was created.

Attribute Name	Description

Type	The ID of the Key Parameter Type that relates to the given value.
Value	The value provided either automatically by the Line of Business system or manually entered by an end user.

Logical Data Viewpoint Model (DIV-2)

The following diagram defines relationships between data elements, providing a structured representation of data flows, entities, and attributes.



Physical Data Viewpoint Model (DIV-3)

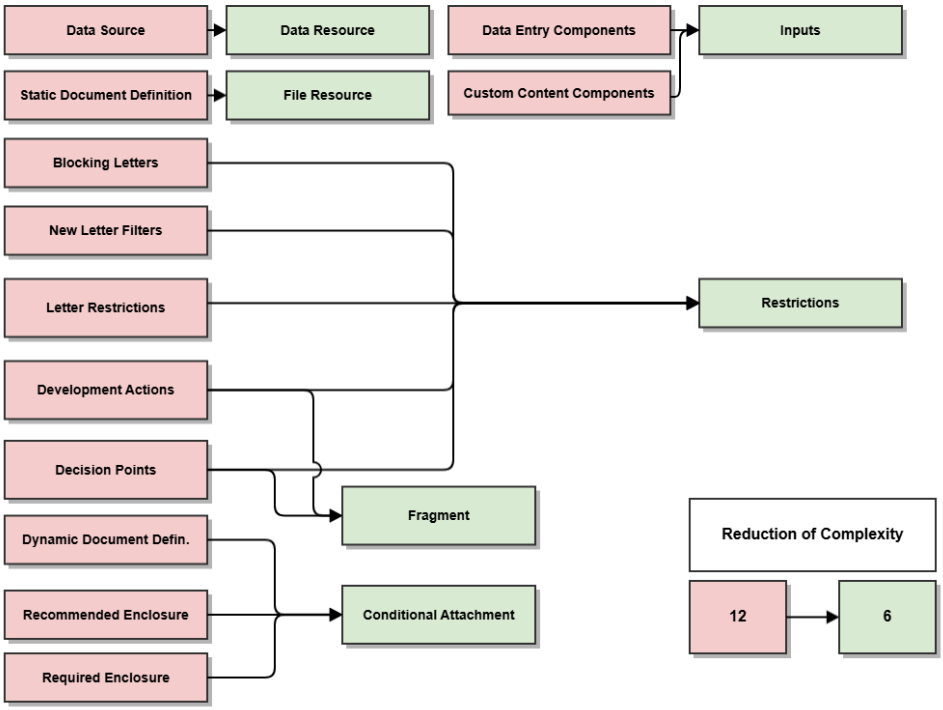
N/A

Transitioning Concepts From Legacy Correspondence Management

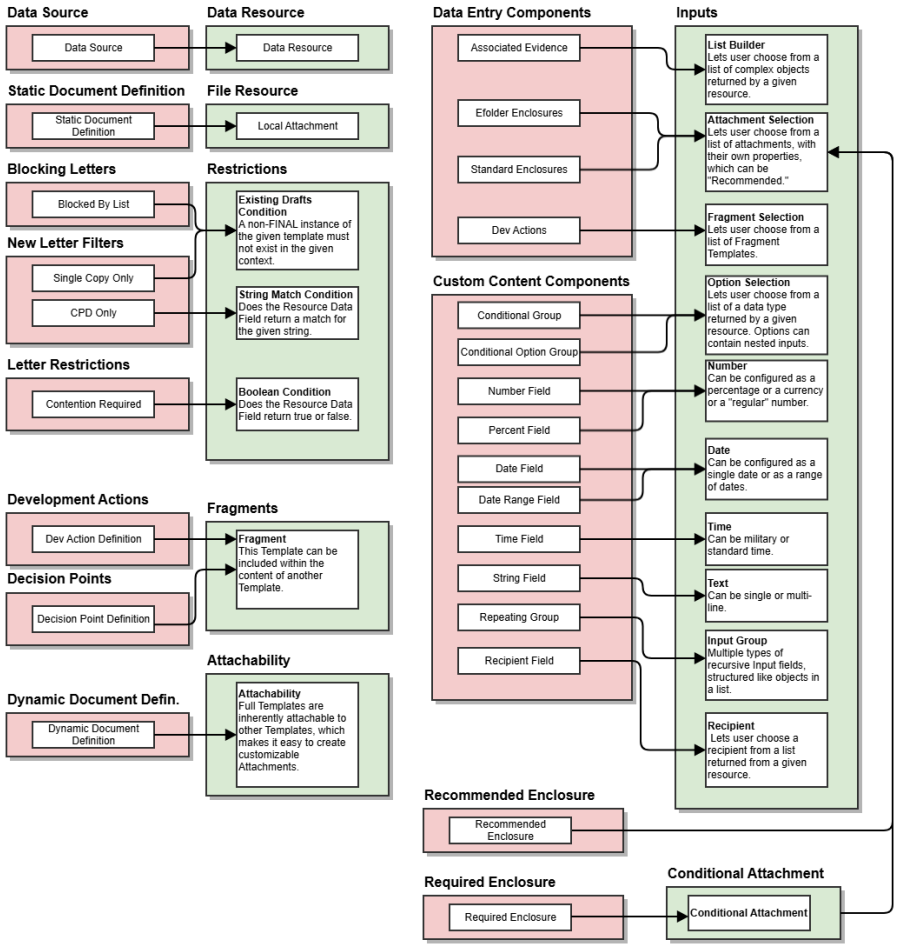
This section visually translates the reimagination of terms from legacy correspondence management applications into ways easier to consume within ECM-L.

ECM-L reduces the complexity and learning curve for new users by consolidating fragmented concepts into elegant, intuitive abstractions that leverage planned architectural complexity behind the scenes, reducing difficult operations to point-and-click activities and transforming technical complexity into a streamlined and simplified user experience.

High Level Concept Changes



Low Level Concept Changes



Claim Evidence and Package Manager Integration

All Lines of Business are encouraged to use Claim Evidence for document storage and Package Manager for correspondence distribution.

What Are Claim Evidence and Package Manager?

Claim Evidence is a system that manages and stores documents related to a specific Veteran and *Package Manager* is a system that allows for the mailing and distribution of the documents stored within Claim Evidence.

Lines of Business are able to configure specific Workflows to use Claim Evidence and / or Package Manager, and if they have opted to use them, they are able to override those configurations for specific types of correspondence.

> **Package Manager Distributions**
>
> Packages created by the ECM-L components are NOT automatically sent, but are ready for review within the Package Manager interface. This is allows for the user to confirm and validate all of the information before sending the packages out to their distribution lists.

Workflow Configuration

Line of Business Administrators are able to configure their Workflows so that they function in specific ways.

Key Parameter Types

Workflows should be configured with one or more Key Parameter Types. These will help shape the list of Letters a user is presented with when they access that Workflow in the Letter Manager.

> *Example:*
>
> *Workflow Name: Developing a Claim*
>
> *Key Parameter Type: C&P Claim ID*
>
> *In this situation the Letter Manager user would be, within this workflow, looking at a list of Letter Instances that relate to a specific C&P Claim ID.*

Storage and Distribution

Users can configure a specific Workflow to use Claim Evidence for document storage. For any Workflows that have been configured to use Claim Evidence, they can optionally configure them to create packages for them in Package Manager.

> Any Workflow that is configured to use Claim Evidence will require the system-level "Participant ID" Key Parameter. In addition to this, each Template must be assigned a Workflow-level Claim Evidence Document Type, and the user will not be able to save a Workflow configured to use Claim Evidence if there are any Templates with an unassigned Claim Evidence Document Type (Unless that Template is marked as Claim Evidence Exempt, which means it doesn't need a Claim Evidence Document Type.)

> **Package Manager Requires Claim Evidence**
>
> You can not use Package Manager without Claim Evidence, because Package Manager only creates packages of documents that are already in Claim Evidence.

Template Association

Workflows have a list of Templates (Templates, not Template Versions), that are associated to them. This represents the list of Templates that a Letter Manager is able to create while within that Workflow.

Each of the Templates in this list can be configured, at the Workflow level, to be Claim Evidence or Package Manager exempt, which would prevent them from being sent to Claim Evidence or Package Manager. These options will appear conditionally, depending on the configuration of the Workflow as a whole.

Document Type

For Workflows configured to use Claim Evidence storage, Templates must be configured with a Claim Evidence Document Type, unless they are marked as Claim Evidence Exempt (see below).

Claim Evidence Exempt

For Workflows configured to use Claim Evidence storage, Templates can be configured as Exempt from Claim Evidence, which means that even though the Workflow is configured to store the PDF of the Finalized Letter in Claim Evidence, that specific Template will not be stored.

Package Manager Exempt

For Workflows configured to use Package Manager distribution, Templates can be configured as Exempt from Package Manager, which means that even though the Workflow is configured to distribute the stored PDF via Package Manager, that specific Template will not be distributed.

Intake

Initial Evaluation

The initial evaluation during the intake process will evaluate how the Line of Business intends to interact with the ECM-L components, or in situations where the Line of Business is migrating from another correspondence management system, will evaluate how it currently interacts with that. Based on this evaluation the ECM-L team can help to outline a strategy that adapts their intended and expected needs, creating a plan for them to interact with the ECM-L framework.

Factors to consider during the evaluation include:

- Integration Points - An evaluation of when and how Lines of Business systems could and should interact with the ECM-L components to provide the best user experience.
  - Are their steps in the process that are not initiated by a user?
  - Do the Lines of Business systems need to process Finalized Letter Data for post-Finalization work?
  - Is there a need to embed the Letter Manage MFE in one of the Line of Business system interfaces?
- Letter Migration - An evaluation of specific pieces of correspondence, providing advice on how to adapt those in terms of the ECM-L Template framework.

- Security and Roles - An evaluation of any Line of Business specific security policy needs.

Line of Business Intake

The intake process for a new Line of Business involves an ECM-Line of Business Intake Form, which outlines how the new Line of Business interacts with ECM-L system and its components, and identifies any Connection Configurations the Line of Business wants to include in the intake process.

Line of Business Intake Form

Enterprise Correspondence Management - Line of Business Intake Form

The Line of Business will submit the Line of Business Intake Form, and Connection Configuration Intake Form if needed, and create a Jira ticket. After submission of the Intake Form(s), there will be an Integration Meeting with stakeholders to introduce ECM-L capabilities, integration points and connection configurations.

Intake Processing

This form ultimately results in the configuration information being loaded in to the Config repo for the Resource Manager where it becomes accessible to the rest of the ECM-L components.

System Registry

As a part of the Intake Process the ECM-L team will determine if the Line of Business will need to interact with the components with a non-human system actor, for processes that are automated. We need to collect enough information to contact Lines of Business to inform them of when Major Changes to a Template they use could necessitate changes to their downstream systems.

Information we'd need to collect from them:

- System Name
- Points of Contact
    - Name
    - Email Address

We will use this registry to alert users to actions that could negatively impact downstream systems so that they can take action to coordinate with the necessary teams to prepare for those changes.

For example, if a new version of a Template is marked as Approved during the Changeset approval process, and that Template is known to be generated by a registered system, the user will be alerted.

Connection Configuration Intake

The connection configuration intake process results in a successful integration between the ECM-L applications and an external Line of Business API. This process begins with a Connection Configuration Intake Form filled out by the Line of Business, which results in a Change Request that loads the necessary Secret / Issuer key pair into the external API and into the individual ECM-L components, and ultimately ends with a Connection Configuration in the Provisioned status, ready to be used.

Connection Configuration Intake Form

Here is an example Change Request ticket that can be cloned in Jira: **BPS-66120** - Getting issue details... STATUS .

The Connection Configuration intake form for a new LoB requesting connection to ECM-L will require the following:

- Line of Business
- Point of Contact
    - Name
    - Email Address
- Connection Name
- Integration Information
    - Vault Secret Locations
    - URL

User Roles and Permissions / Intake

ECM-L has a robust and well organized system for dealing with Line of Business specific user permissions and roles.

For more information, see: ECM-L User Accounts and Permissions.

Process View (Business Process Model)

The Process View describes the behavior of the systems, the sequence of events, and how various tasks combine to the functionality described in the Use-Case View. It decomposes the systems into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). The Process View describes how systems interact, detailing the timing of messages or events.

Main Business Process Model (OV-6d)

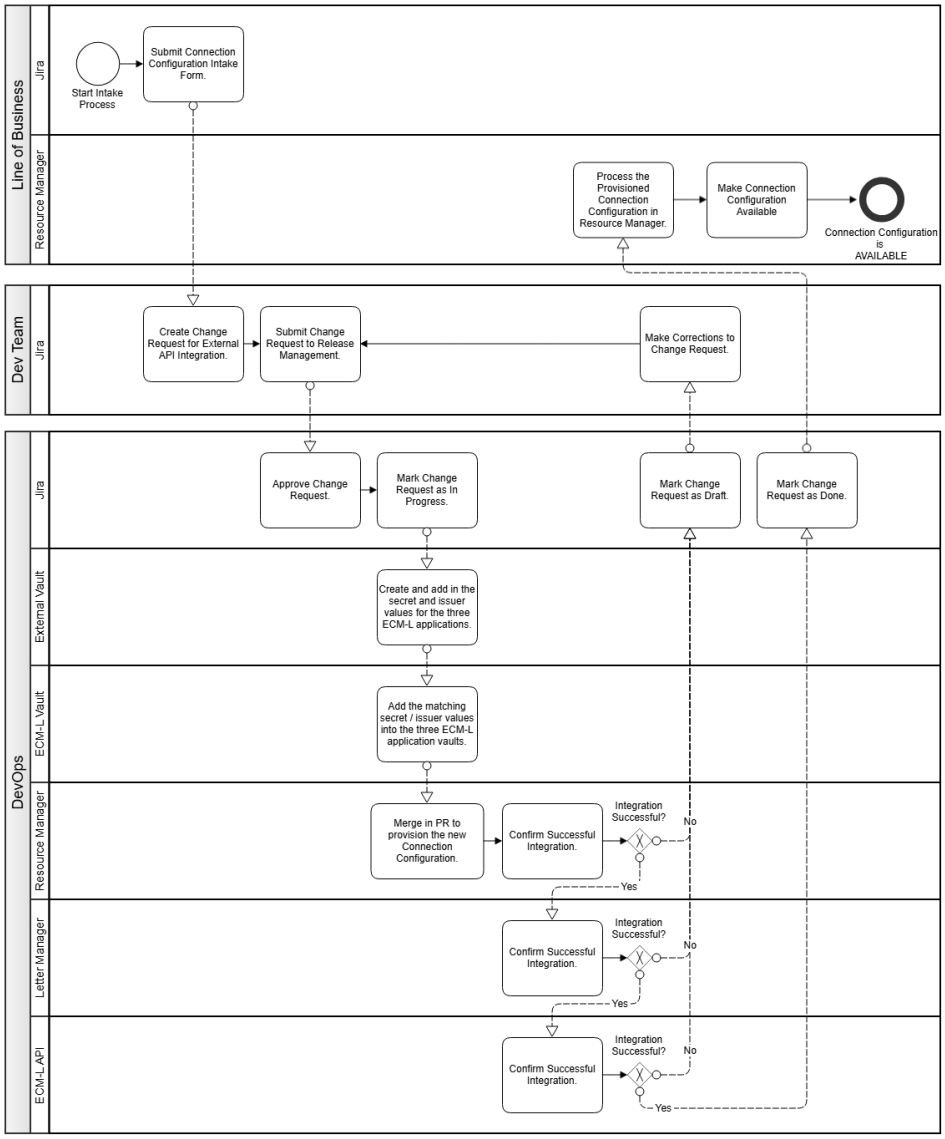
Intake Processes

The following diagrams delineate the key processes and workflows within ECM-L, mapping the sequence of activities and information flow that drive operations.

Connection Configuration Process Workflows

Process Workflow for Connection Configuration Intake

This process guides the intake and integration of new connections. It includes submission, approval, configuration, and confirmation across multiple systems.

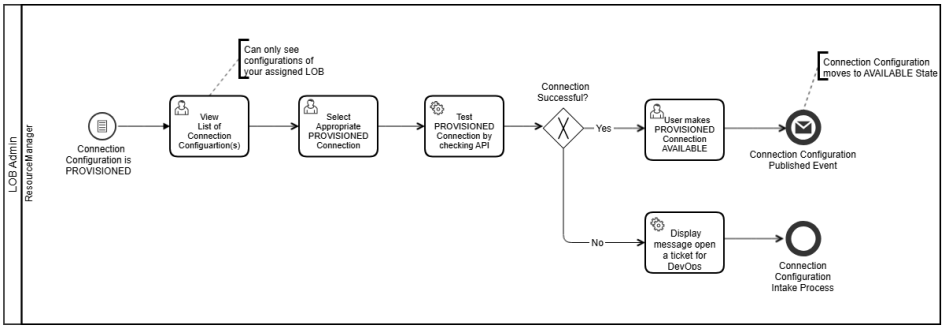


Task	Description
Start Intake Process	Initiate the process for a new connection configuration request.

Submit Connection Configuration Intake Form	Resource Manager submits the intake form to begin provisioning.
Create Change Request for External API Integration	Dev Team creates a change request for required API integration.
Submit Change Request to Release Management	Dev Team submits the change request for approval and release tracking.
Approve Change Request	Release Manager reviews and approves the submitted change request.
Mark Change Request as In Progress	Release Manager (or Dev Team) marks change request as actively being worked on.
Make Corrections to Change Request	Dev Team revises the change request as needed based on feedback or errors.
Create and add secret/issuer values (ECM Vault)	External Vault team generates and adds secret/issuer values for ECM applications.
Add secret/issuer values to ECM application vaults	ECM Vault team updates ECM application vaults with the required secret/issuer values.
Merge in PR to provision new Connection Config	Resource Manager merges pull request to provision the connection configuration.
Confirm Successful Integration	Various teams (Resource Manager, Ledger Manager, ECM API) confirm that the integration was completed successfully.
Integration Successful?	Confirmation step checks if integration was successful; if not, process returns to correction steps.
Make Connection Configuration Available	Once integration is successful, configuration is made available for use.
Connection Configuration is AVAILABLE	The configuration is fully provisioned and available for business needs.

Make Connection Available

The Line of Business Administrator is able to make a Connection Configuration available.

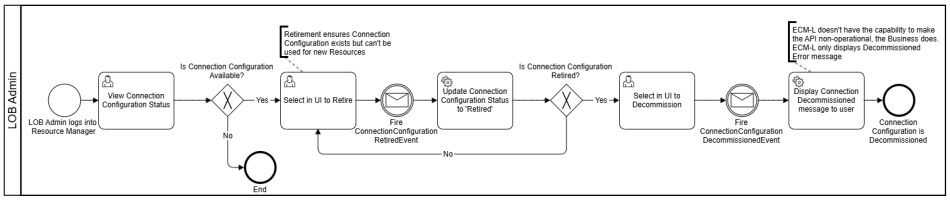


Task	Description
View List of Connection Configuration(s)	LOB Admin views available connection configurations, filtered by their assigned line of business.
Select Appropriate PROVISIONED Connection	LOB Admin selects the provisioned connection to be made available.
Test PROVISIONED Connection by checking API	LOB Admin tests the selected connection by verifying its API functionality.
Connection Successful?	Decision point to confirm if the connection test was successful.
User makes PROVISIONED Connection AVAILABLE	If successful, the admin publishes the connection, making it available.
Connection Configuration Published Event	System records and completes the publication event, moving the connection to AVAILABLE state.
Display message open a ticket for DevOps	If unsuccessful, system prompts user to open a support ticket for DevOps intervention.
Connection Configuration Intake Process	Process returns to intake for troubleshooting and resolution if integration fails.

Connection Configuration Decommissioning Process

The process of Decommissioning a Connection Configuration starts with the LOB Admin Retiring the Connection Configuration first and then marking it as 'Decommission' in the Resource Manager. Retirement ensures that the Connection Configuration can not be used to create new Resources.

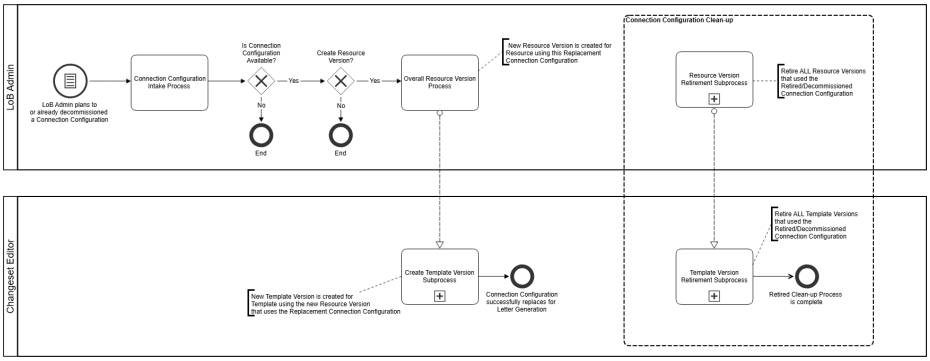
After the Connection Configuration is Retired it can be Decommissioned which will only show a message that the API is non-operational. The actual Decommissioning happens outside of ECML.



Task	Description
LOB Admin Login	The LOB Admin logs in to the system to manage Connection Configurations.
View Connection Configuration Status	The LOB Admin views the status of available Connection Configurations.
Is Connection Configuration Available?	Decision Point: Is the Connection Configuration available for retirement?
If No End	If no Connection Configuration is available, the process ends.
If Yes Select in UI to Retire	The LOB Admin selects the Connection Configuration in the UI to retire it.
Fire ConnectionConfigurationRetiredEvent	The system fires the ConnectionConfigurationRetiredEvent to initiate retirement.
Update Connection Configuration Status to 'Retired'	The Connection Configuration status is updated to 'Retired'.
Is Connection Configuration Retired?	Decision Point: Has the Connection Configuration been successfully retired?
If No Select in UI to Retire	If not retired, the process reverts back to the User to select in the UI to Retire.
If Yes Select in UI to Decommission	The LOB Admin selects the retired Connection Configuration in the UI to decommission it.
Fire ConnectionConfigurationDecommissionedEvent	The system fires the ConnectionConfigurationDecommissionedEvent to initiate decommissioning.
Display Connection Decommissioned message to user	The system displays a message to the user confirming the Connection Configuration is decommissioned.
Connection Configuration is Decommissioned	The Connection Configuration is fully decommissioned and removed from use for new resources.

Connection Configuration Replacement Process

The model demonstrates the best practice for replacement of a Connection Configuration that has been or is planned on being Decommissioned. This includes replacing the Connection Configuration's Template Versions and Resource Versions by creating new Versions of each as desired. Additionally, all Resource and Templates Versions that used the old Connection Configuration should be Retired as they are no longer needed.

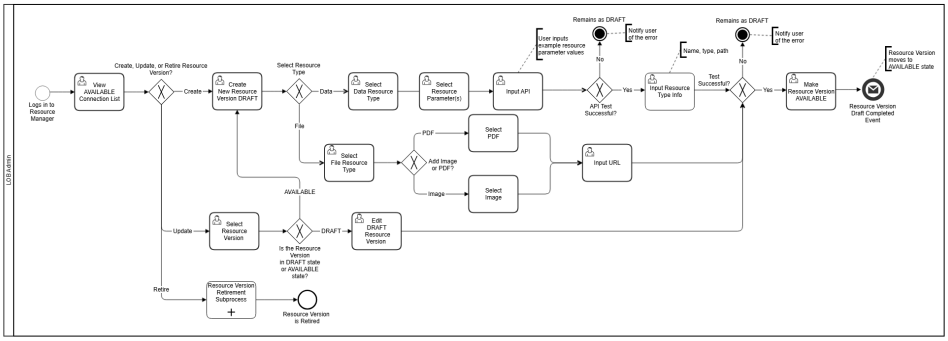


Task	Description

LoB Admin plans to or already decommissioned a Connection Configuration	The LoB Admin identifies a Connection Configuration that is planned for or already decommissioned.
Connection Configuration Intake Process	The LoB Admin initiates the intake process to evaluate and process the replacement Connection Configuration.
Is Connection Configuration Available?	Decision Point: Determine if a replacement Connection Configuration is available.
If No End	If no replacement is available, the process ends.
Create Resource Version?	Decision Point: Determine if a new Resource Version should be created using the replacement Connection Configuration.
If No End	If not creating a Resource Version, the process ends.
Overall Resource Version Process	If yes, initiate the process to create a new Resource Version using the replacement Connection Configuration.
New Resource Version is created for Resource using this Replacement Connection Configuration	A new Resource Version is created using the replacement Connection Configuration.
Create Template Version Subprocess	The Changeset Editor initiates the subprocess to create a new Template Version using the new Resource Version.
New Template Version is created for Template using the new Resource Version	A new Template Version is created for the Template that uses the replacement Connection Configuration.
Connection Configuration successfully replaces for Letter Generation	The new Connection Configuration is now successfully used for letter generation.
Connection Configuration Clean-up	Initiate clean-up process for the retired/decommissioned Connection Configuration.
Resource Version Retirement Subprocess	Retire ALL Resource Versions that used the retired/decommissioned Connection Configuration.
Template Version Retirement Subprocess	Retire ALL Template Versions that used the retired/decommissioned Connection Configuration.
Retired Clean-up Process is complete	The process of cleaning up retired Resource and Template Versions is complete.

Overall Resource Version Process

The model demonstrates the creation, update and retirement of a Resource Version within Resource Manager. Two types of Resources can be added, each requiring testing before they become available for use.



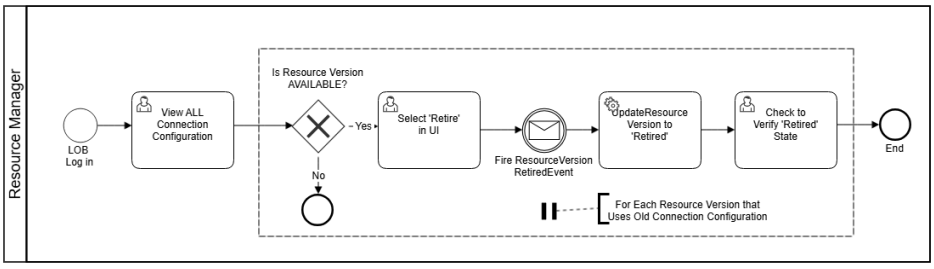
Task	Description
Log in to Resource Manager	The LOB Admin logs into the Resource Manager to manage resource versions.
View AVAILABLE Connection List	The LOB Admin views the list of available connections for resource creation, update, or retirement.
Create, Update, or Retire Resource Version?	Decision Point: Determine if the action is to create, update, or retire a resource version.
Create New Resource Version DRAFT	If creating, the LOB Admin initiates a new resource version in DRAFT state.
Select Resource Type	The LOB Admin selects the type of resource to create (Data or File).
Select Data Resource Type	For data resources, select the specific data resource type.
Select Resource Parameter(s)	Input the parameters for the data resource version.

Input API	Enter API details if required for the resource type.
API Test Successful?	Decision Point: Test the API with sample values to confirm it works as expected.
If No Remains as DRAFT	If the API test is unsuccessful, the resource version remains in DRAFT and the user is notified of the error.
If Yes Input Resource Type Info	If the API test is successful, input resource type information (name, type, path).
Test Successful?	Decision Point: Test resource type information for success.
If No Remains as DRAFT	If not successful, the resource version remains in DRAFT and the user is notified of the error.
If Yes Make Resource Version AVAILABLE	If successful, make the resource version AVAILABLE.
Resource Version Draft Completed Event	The system triggers the Resource Version Draft Completed Event and moves the resource version to AVAILABLE state.
Select File Resource Type	For file resources, select the specific file resource type.
Add Image or PDF?	Decision Point: Determine if the file resource is an image or PDF.
Select PDF	If PDF, select the PDF file to add.
Select Image	If image, select the image file to add.
Input URL	Enter the URL for the file resource if required.
Update	For updates, select the resource version to modify.
Is the Resource Version in DRAFT or AVAILABLE state?	Decision Point: Is the resource version in DRAFT or AVAILABLE state?
Edit DRAFT Resource Version	If DRAFT, edit the resource version details.
Resource Version Retirement Subprocess	If retiring, initiate the Resource Version Retirement Subprocess.
Resource Version is Retired	The resource version is retired and no longer available for use.

Resource Version Retirement Subprocess

This model demonstrates LOB admin viewing list of Resource Versions for a Connection Configuration and retiring each Resource Version which updates the Resource Version to RETIRED state.

Retiring the Resource Version ensures no new Resources can be created using this retired Connection Configuration.

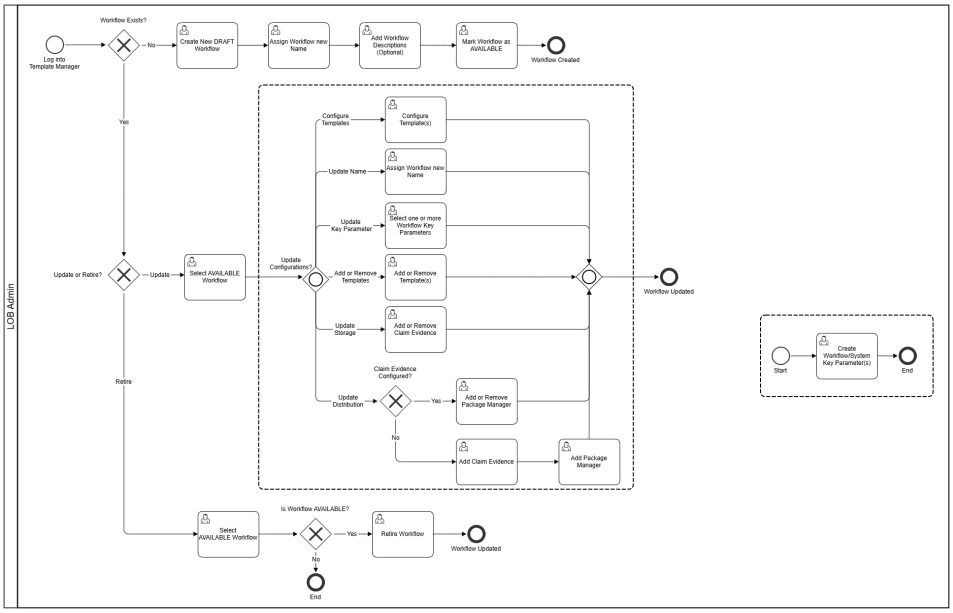


Task	Description
LOB Log in	The Line of Business (LOB) user logs in to the system.
View ALL Connection Configuration	The Resource Manager views all available connection configurations.
Is Resource Version AVAILABLE?	Decision Point: Is the Resource Version available for retirement?
If No End	If the Resource Version is not available, the subprocess ends.
If Yes Select 'Retire' in UI	The Resource Manager selects the 'Retire' option for the Resource Version in the UI.
Fire ResourceVersionRetiredEvent	The system fires the ResourceVersionRetiredEvent to initiate retirement of the Resource Version.
Update Resource Version to 'Retired'	The system updates the status of the Resource Version to 'Retired'.
Check to Verify 'Retired' State	The Resource Manager checks to verify that the Resource Version is in 'Retired' state.

For Each Resource Version that Uses Old Connection Configuration	Repeat the subprocess for each Resource Version that uses the old Connection Configuration.
End	The Resource Version Retirement subprocess is complete.

Create / Update / Retire Workflow

This model demonstrates how workflow is created, updated with additional configurations and retired from its AVAILABLE state by LOB Admin. Additional configurations include adding or removing Full Templates and storage and distribution options with Claim Evidence and Package Manager. Claim Evidence is responsible for the storage of Documents and Package Manager accepts request from Claim Evidence to distribute correspondence to the recipient.



Task	Description
Log in to Template Manager	The LoB Admin logs into Template Manager.
Workflow Exists?	Decision Point: The LoB Admin checks to see if the Workflow already exists.
No Create New DRAFT Workflow	If a Workflow does not exist, the LoB Admin creates a new Workflow in the DRAFT state.
Assign Workflow new Name	The LoB Admin assigns a new Name to the Workflow.
Add Workflow Description (Optional)	The LoB Admin adds an optional description for the Workflow
Mark Workflow as AVAILABLE	LoB Admin marks the Workflow as AVAILABLE.
Workflow Created	Workflow is created.
Workflow Exists? Yes Update or Retire?	The LoB Admin can choose to Update or Retire the Workflow.
Update Select AVAILABLE Workflow	The LoB Admin selects the AVAILABLE Workflow to update the Configurations.
Select Configurations? (options)	The LoB Admin determines the Configurations they want to update.
Configure Templates	The LoB Admin selects to configure Templates already added to the Workflow. Here they are able to make the Template, at the Workflow level, Package Manager or Claim Evidence Exempt, and configure it with a Claim Evidence Document Type as needed.
Update Name	The LoB Admin selects to updates the Workflow Name.
Update Key Parameter	The LoB Admin selects to updates the Workflow Parameter.
Add or Remove Template(s)	The LoB Admin selects to Add or Remove Templates.

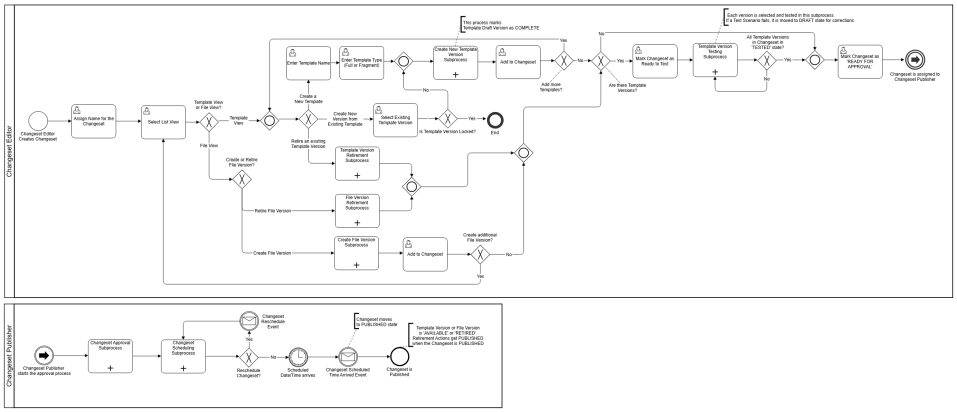
Update Storage	The LoB Admin selects to update Storage.
Update Distribution	The LoB Admin selects to update Distribution.
Update Name Assign Workflow new Name	The LoB Admin assigns a new Name to the Workflow.
Update Key Parameter–> Select one or more Workflow Key Parameters	The LoB Admin selects one or more Workflow Key Parameters to add to the Workflow.
Manage Template–> Add or Remove Template	The LoB Admin adds or removes a Template.
Update Storage Add or remove Claim Evidence	The LoB Admin adds or removes Claim Evidence as Storage.
Update Distribution Claim Evidence Configured?	Decision Point: Is the Workflow configured with Claim Evidence?
Claim Evidence Configured? Yes Add or remove Package Manager	The LoB Admin adds or removes Package Manager for Distribution.
Claim Evidence Configured? No Add Claim Evidence	The LoB Admin configures Claim Evidence for Storage
Add Claim Evidence Add Package Manager	The LoB Admin configures Package Manager for Distribution.
Workflow Updated	After all desired Configuration Updates are performed, the Workflow is updated.
Retire Select Workflow	The LoB Admin selects the Workflow to be retired.
Is Workflow AVAILABLE?	Is the selected Workflow AVAILABLE?
If Yes Retire Workflow	The LoB Admin retires the Workflow.
If No Workflow is not Retired	Workflow cannot be retired.

Overall Changeset Process

This model outlines the complete lifecycle of a Changeset from creation to publishing. It illustrates key states, decision points, and transitions across testing, approval, and scheduling.

Changeset Editor Swimlane demonstrates the Changeset Editor creates a Changeset first and creates a new Template Version, add an existing Template Version or retire a Template Version. Once all the Template Versions are in COMPLETED state, each Template Version can be tested in the Changeset. Changeset can be submitted for approval when all the Template Versions pass the test scenarios and their status are updated to TESTED. Changeset Editor also creates a new File Version or retires a File Version. Changeset Editor marks the File Version as COMPLETE to submit for approval.

Changeset Publisher Swimlane demonstrates Changeset Publisher can approve or reject the Changeset. When the Changeset is approved, it can be scheduled or rescheduled for publishing. Upon publishing, the Changeset Status is updated to PUBLISHED.



Overall Changeset Process

Changeset Editor

Task	Description

Changeset Editor Creates Changeset	Initiate the process by creating a new changeset to manage template and/or file version changes.
Assign Name for the Changeset	Give the changeset a unique, descriptive name for tracking and identification.
Select List View	Choose the view mode for managing either template versions or file versions.
Decision: Template View or File View?	Decide whether the next action involves templates or file versions.
Decision: Create a New Template / Existing?	In template view, decide to either create a new template or work with an existing template version.
Enter Template Name	If creating a new template, specify its name.
Enter Template Type	Select whether the Template Type is a Full or Fragment Template Type.
Create New Template Version Subprocess	Begin the subprocess to create a new version for a template, entering all required information and details.
Add to Changeset (Template)	Add the newly created template version to the changeset for further processing.
Decision: Add More Templates?	Decide whether to add additional template versions to the changeset or proceed.
Select Existing Template Version	If not creating a new template, select an existing template version for update or retirement.
Decision: Is Template Version Locked?	Check if the selected template version is locked; if locked, the update process cannot proceed and ends for that item.
Create New Version from Existing Template	If not locked, initiate the creation of a new version from the existing template.
Template Version Retirement Subprocess	If retiring, start the subprocess to retire the selected template version.
Add to Changeset (from Retirement /Update)	Add template versions (created or retired) to the changeset.
Decision: Are there Template Versions?	Decide if any template versions are present in the changeset to proceed to testing.
Mark Changeset as Ready to Test	If template versions are present, mark the changeset as ready to begin the testing process.
Template Version Testing Subprocess	Initiate the subprocess to test each template version in the changeset. Failed scenarios move items back to DRAFT for correction.
Decision: All Template Versions in TESTED?	Check if all template versions in the changeset have passed testing and reached the TESTED state.
Submit Changeset for Approval	Once all template versions are tested, submit the changeset for approval by the publisher.
File View	If file view is selected, manage file versions within the changeset.
Decision: Create or Retire File Version?	Decide whether to create a new file version or retire an existing file version.
Create File Version Subprocess	Start the subprocess to create a new file version, including versioning, naming, and validation.
File Version Retirement Subprocess	Begin the subprocess to retire an existing file version.
Add to Changeset (File)	Add created or retired file versions to the changeset for processing.
Decision: Create Additional File Version?	Decide whether to create or retire additional file versions or proceed to the next steps in the process.

Changeset Publisher

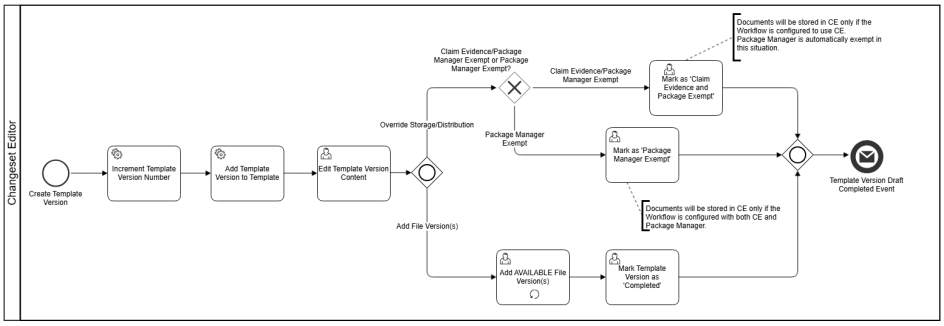
Task	Description
Changeset Approval Subprocess	Review the changeset and approve or reject it.
Changeset Scheduling Subprocess	Schedule the approved changeset for publication.
Changeset Reschedule Event	Optionally reschedule the publication of the changeset.
Scheduled Date/Time Arrives	Monitor until the scheduled date and time for publication arrives.
Changeset Scheduled Time Arrived Event	Trigger publication when the scheduled time is reached.
Changeset is Published	Finalize and publish the changeset.

Changeset moves to PUBLISHED state	Update the status of the changeset, template versions, and file versions to AVAILABLE or RETIRED as appropriate.

Create Template Version for Changeset Subprocess

This model demonstrates the Changeset Editor creating a Template Version and editing its content as needed. Changeset Editor adds the desired File Version(s) to the Template Version and marks it as COMPLETE so the status of the Template Version updates to COMPLETED state.

The Changeset Editor configures the Template Version by either following the Workflow's existing settings or overriding them for storage and distribution. The Changeset Editor can choose to bypass both Claim Evidence and Package Manager, or use Claim Evidence only by exempting Package Manager.



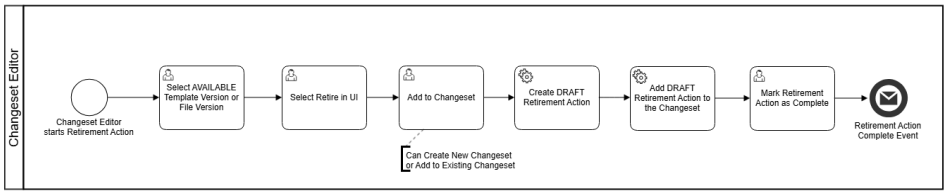
Task	Description
Create Template Version	The process begins with Changeset Editor initiating the creation of a Template Version.
Increment Template Version Number	The system increases the version number for the Template to reflect the new Version.
Add Template Version to Template	The system adds the newly created Template Version to the collection of Templates in the system.
Edit Template Version Content	The Changeset Editor edits and refines the content of the DRAFT Template Version as needed.
Add File Version(s) Add AVAILABLE File Version(s)	The Changeset Editor wants to add one or more File Version(s) in the AVAILABLE state to the DRAFT Template Version.
Mark Template Version as 'Completed'	The Changeset Editor marks the Template Version as 'Completed'.
Override Storage/Distribution Claim Evidence/Package Manager Exempt or Package Manager Exempt?	The Changeset Editor wants to override the storage/distribution.
Claim Evidence/Package Manager Exempt or Package Manager Exempt?	Decision Point: The Changeset Editor can bypass both Claim Evidence and Package Manager or only Package Manager.
If Claim Evidence/Package Manager Exempt Mark as 'Claim Evidence and Package Manager Exempt'	The Changeset Editor marks the Template Version as 'Claim Evidence and Package Manager Exempt. Documents will be stored in CE only if the Workflow is configured to use CE. Package Manager is automatically exempt in this situation.
If Package Manager Exempt Mark as 'Package Manager Exempt'	The Changeset Editor marks the Template Version as 'Package Manager Exempt'. Documents will be stored in CE only if the Workflow is configured with both CE and Package Manager.
Template Version Draft Complete Event	The Template Draft Completed Event is triggered. **Template Version moves to COMPLETED state*

Template Version and File Version Retirement Subprocess

Template Version and File Version retirement is a stand-alone process, but also integral to the Connection Configuration Replacement process and Overall Changeset process.

This model demonstrates Changeset Editor selecting an AVAILABLE Template Version or File Version, and invoking the Retirement Action, which changes its status to RETIRED and remove it from active use.

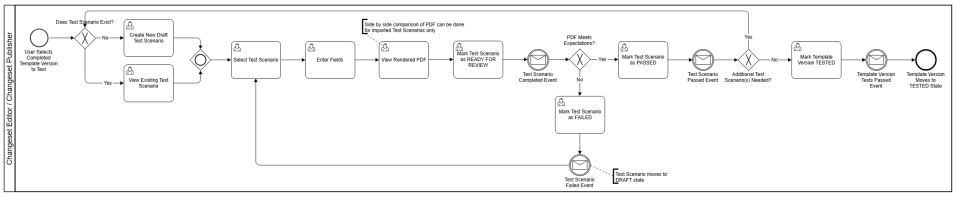
> To Retire all Versions of a Template or File at once, the Changeset Editor can create a Changeset with Retirement Actions only.



Task	Description
Changeset Editor starts Retirement Action	The Changeset Editor logs in to Template Manager to start the Retirement Action
User Selects AVAILABLE Template Version(s) or File Version(s)	Changeset Editor selects AVAILABLE Template Version(s) or File Version(s) to retire.
User Selects 'Retire' in UI	The Changeset Editor selects to retire the Template Version.
Add to Changeset	The Changeset Editor adds the Template Version(s) or File Version(s) to the Changeset
Create DRAFT Retirement Action	Template Manager creates a DRAFT Retirement Action containing the Template Version(s) or File Version(s) selected to be retired.
Add DRAFT Retirement Action to the Changeset	Template Manager adds the Retirement Action to the Changeset.
Mark Retirement Action as Complete	The Changeset Editor marks the Retirement Action as Complete.
Retirement Action Complete Event	The Retirement Action Complete Event is fire, and the Retirement Action moves to COMPLETED state.

Template Version Testing Subprocess

This model demonstrates how the Changeset Editor creates new or selects existing test scenario to validate Template Versions, ensuring they render the correct PDF with appropriate changes. Upon successful completion of all test scenarios, the Changeset Editor marks them as TESTED, which transitions the Changeset to a TESTED state.
Test Scenarios are created for specific Template Versions, and are used inside of Changesets to test them, but live outside of the Changeset, and can be re-used by future Changesets to regression test changes to those Templates.

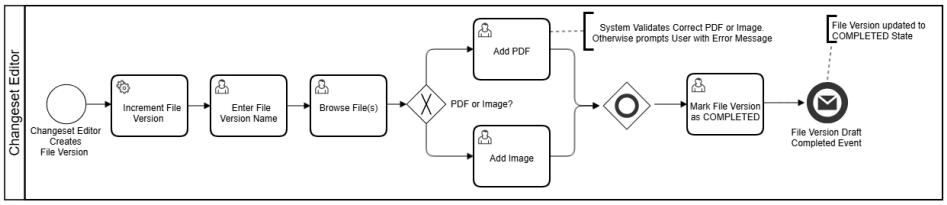


Task	Description
User COMPLETED Template Version to Test	The process starts when the Changeset Editor has a COMPLETED Template Version that they want to test.
Do Test Scenarios for the Template Version Exist?	Decision Point: determine if Test Scenario(s) already exist for the Template Version.
If No Create New Test Scenario for Template Version	If Test Scenario(s) do not already exist, the Changeset Editor creates a new Test Scenario for the Template Version.
If Yes View Existing Test Scenarios for Template Version	If Test Scenario(s) exist, the Changeset Editor view the existing Test Scenario (s) for the Template Version.
Select Test Scenario	The Changeset Editor selects the appropriate Test Scenario to work with.
Enter Fields in Selected Test Scenario	The Changeset Editor enters the required fields for the selected Test Scenario.
If New Test Scenario Show Rendered Sample PDF	If the Test Scenario is new, the system renders and displays the sample PDF.
If Existing Test Scenario Show Side by Side Comparison Between PDF Renders of This Version and Last Version	If there are existing Test Scenario(s), the system shows a side-by-side comparison of current and previous PDF Template Versions.
PDF Meet Expectations	Decision point: do the results of the PDF meet the Changeset Editor's expectations?
If Yes Mark Test Scenario as Passed	If yes, mark the Test Scenario as PASSED.

Test Scenario Passed Event	The system records the Template Version Tests Passed event.
Are Additional Test Scenario(s) needed?	Decision Point: Determine if additional testing is needed.
If Yes Go back to check if Test Scenario(s) exist	If yes, go back to check if the Test Scenario exists and continue through the steps.
If No Mark Template Version Tested	If no additional Test Scenario(s) are needed, the Changeset Editor marks the Template Version as TESTED.
Template Version Tests Passed Event	The system records the Template Version Tests Passed event.
Template Version Moves to Tested State	The Template Version moves to the TESTED state.
If No Mark Test Scenario as Failed	If no, mark the Test Scenario as FAILED. The Changeset Editor loops back for further action.
Test Scenario Failed Event	The system records the Test Scenario Failed event and moves the Test Scenario to DRAFT state.

Create File Version Subprocess

The Create File Version Subprocess allows a Changeset Editor to create and prepare a new file version for inclusion in a changeset. The process includes versioning, naming, file selection and validation, and setting the file version to a completed state once requirements are met.

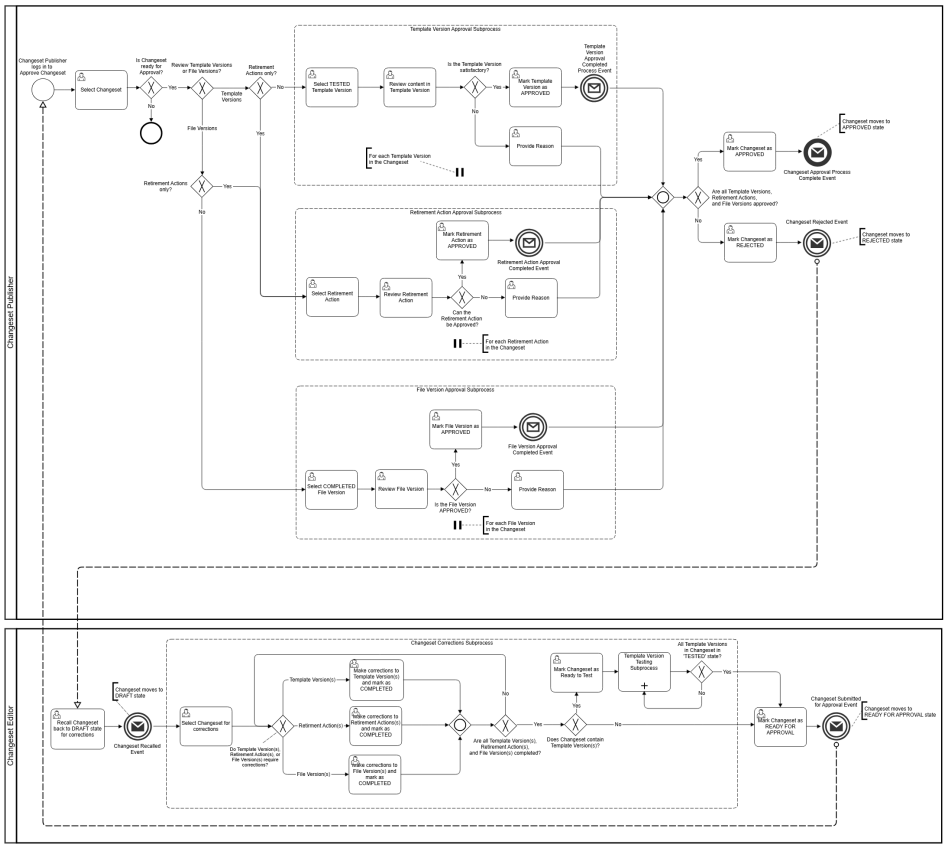


Task	Description
Increment File Version	The system increments the file version number for a new file version entry.
Enter File Version Name	The editor enters a unique name for the new file version.
Browse File(s)	The editor browses and selects the file(s) to be added (PDF or image).
Add PDF / Add Image	The editor uploads either a PDF or an image, as determined by file selection.
Validate File Type	The system validates the uploaded file is a correct PDF or image; errors prompt user action.
Mark File Version as COMPLETED	Once validation passes, the editor marks the file version as completed.
File Version Draft Completed Event	The file version is updated to the COMPLETED state and is ready for further workflow steps.

Changeset Approval Subprocess

This model demonstrates the Changeset Publisher reviewing the TESTED Template Versions, the Retirement Actions and File Versions within the Changeset to approve or reject them. Once all the Template Versions, Retirement Actions and File Versions are approved, the Changeset Publisher approves the entire Changeset so that the status of the Changeset updates to APPROVED state.

Changeset Publisher rejects the Changeset if the Publisher doesn't approve the changes for the Template Versions, Retirement Actions or File Versions. The Changeset get's recalled back to the DRAFT state and corrections are made by the Changeset Editor before submitting for approval again.



Changeset Approval Subprocess

Changeset Publisher

Task	Description
Select Changeset	Publisher selects the changeset to review and approve.
Is Changeset Approval Required?	Decision point to determine if approval is needed for the changeset.
Review Template Versions	If template versions exist, publisher reviews each one in the changeset.
Review File Versions	If file versions exist, publisher reviews each one in the changeset.
Review Retirement Actions	If the changeset contains retirement actions, publisher reviews each one.
Mark Changeset as APPROVED	If all versions/actions are approved, publisher marks the changeset as APPROVED.
Mark Changeset as REJECTED	If any version/action is not approved, publisher marks the changeset as REJECTED and provides a reason.
Changeset Approval Process Complete Event	Publisher completes the approval process, transitioning the changeset to the APPROVED state.
Changeset Rejected Event	Publisher completes the rejection process, transitioning the changeset to the REJECTED state.
Are all Template Versions, File Versions, and Retirement Actions approved?	Decision to confirm full approval before moving to final state.

Template Version Approval Subprocess

Task	Description
Select Template Version	Publisher selects each template version in the changeset for review.
Review content in Template Version	Publisher reviews the content and details of the template version.

Is the Template Version APPROVED?	Decision point to approve or reject the template version.
Mark Template Version as APPROVED	If approved, publisher marks the template version as APPROVED.
Provide Reason	If rejected, publisher provides a reason for rejection.
Template Version Approval Completed Event	Publisher completes approval/rejection for each template version.

Retirement Action Approval Subprocess

Task	Description
Select Retirement Action	Publisher selects each retirement action in the changeset for review.
Review Retirement Action	Publisher reviews the details of the retirement action.
Is the Retirement Action APPROVED?	Decision point to approve or reject the retirement action.
Mark Retirement Action as APPROVED	If approved, publisher marks the retirement action as APPROVED.
Provide Reason	If rejected, publisher provides a reason for rejection.
Retirement Action Approval Completed Event	Publisher completes approval/rejection for each retirement action.

File Version Approval Subprocess

Task	Description
Select COMPLETED File Version	Publisher selects each file version in the changeset for review.
Review File Version	Publisher reviews the details and content of the file version.
Is the File Version APPROVED?	Decision point to approve or reject the file version.
Mark File Version as APPROVED	If approved, publisher marks the file version as APPROVED.
Provide Reason	If rejected, publisher provides a reason for rejection.
File Version Approval Completed Event	Publisher completes approval/rejection for each file version.

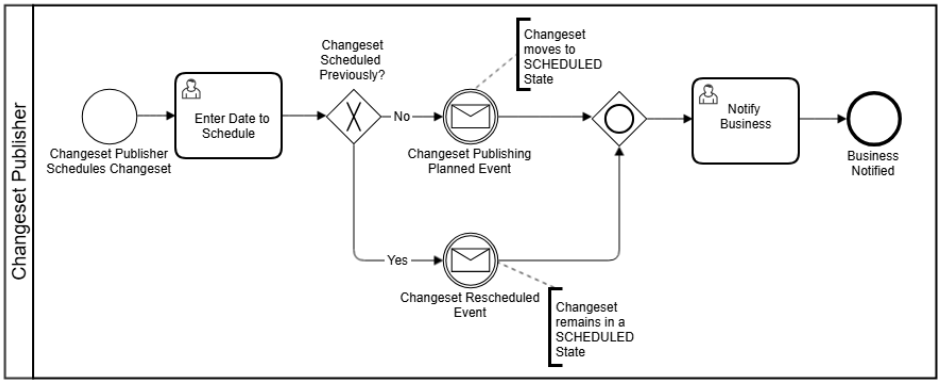
Changeset Editor

Task	Description
Recall Changeset to DRAFT state for corrections	Editor recalls the rejected changeset for modifications.
Select Changeset for corrections	Editor selects the recalled changeset to begin making necessary corrections.
Make corrections to Template Versions	Editor modifies or updates template versions as required.
Make corrections to Retirement Actions	Editor modifies or updates retirement actions as required.
Make corrections to File Versions	Editor modifies or updates file versions as required.
Are all Template Versions and File Versions complete?	Decision to check if everything is updated and complete.
Mark Changeset as Ready to Test	If complete, editor marks the changeset as ready for testing.
Template Version Testing Subprocess	Editor initiates testing for updated template versions.
Are all Template Versions in TESTED state?	Decision to check if all template versions have passed testing.
Changeset contains Template Versions?	Decision to check if the changeset has template versions.
Changeset Submitted for Approval Event	Editor submits the finalized changeset for approval.
Changeset moves to READY FOR APPROVAL state	Changeset status is updated to indicate it is ready for publisher review.

Changeset Scheduling Subprocess

This model demonstrates the Changeset Publisher scheduling the Changeset for publication and notifying business stakeholders of the scheduled date.

Should rescheduling be necessary, the Changeset Publisher can update the publication date accordingly.



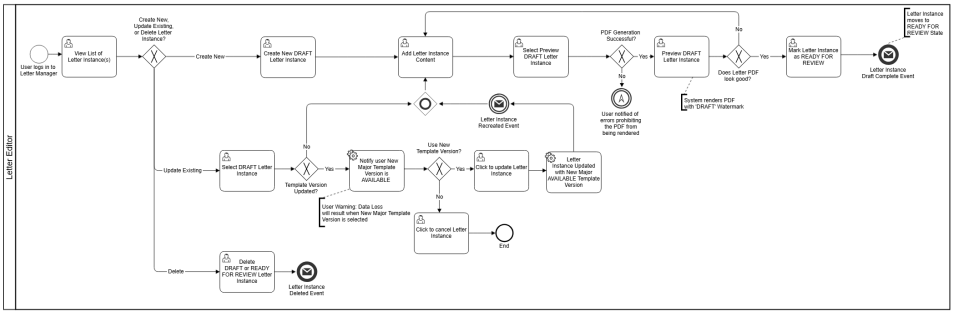
Task	Description
Changeset Publisher Schedules Changeset	The Changeset Publisher selects an APPROVED Changeset to schedule for publishing.
Enter Date to Schedule	The Changeset Publisher enters the desired date/time to schedule the Changeset.
Changeset Scheduled Previously?	Decision Point: Has the Changeset been scheduled previously?
If No Changeset Publishing Planned Event	If not previously scheduled, the Changeset Publishing Planned Event is triggered and the Changeset moves to SCHEDULED state.
If Yes Changeset Rescheduled Event	If already scheduled, a Changeset Rescheduled Event is triggered; Changeset remains in SCHEDULED state.
Notify Business	The business stakeholders are notified of the scheduled date/time for publishing the Changeset.
Business Notified	The notification process is complete; business stakeholders have been informed.

Letter Finalization Process

This model illustrates how Letters are reviewed, approved and Finalized.

The Letter Editor Swimlane demonstrates the Letter Editor's capabilities for managing Letter Instances - creating new ones, modifying existing ones, and deleting as needed. New letters automatically utilize the most recent AVAILABLE Template Version to generate DRAFT PDF, while edits to existing letters prompt users with an option to upgrade to newer Templates or maintain their current version. Once the Letter Editor completes their work and marks the letter as READY FOR REVIEW, the system updates its status to READY FOR REVIEW state.

The Letter Finalizer Swimlane demonstrates the Letter Finalizer reviewing the Letter marked as READY FOR REVIEW to finalize it so the Letter is updated to FINALIZED status. Rejected Letters revert to DRAFT state for revision.



Letter Editor

Task	Description
User logs in to Letter Manager	Letter Editor logs into the Letter Manager to manage Letter Instance(s).
View List of Letter Instance(s)	Displays Letter Instance(s), highlighting those with retired or outdated Template Versions.
Create New, Update Existing, or Delete Letter Instance?	Decision Point: Choose to create, update, or delete a Letter Instance.
Create New Create New DRAFT Letter Instance	Create a new draft Letter Instance using the most current AVAILABLE Template Version.
Update Existing Select DRAFT Letter Instance	Select existing draft Letter Instance to update.
Delete Delete DRAFT or READY FOR REVIEW Letter Instance	Delete draft or ready for review Letter Instance. Triggers Letter Instance Deleted Event.
Add Letter Instance Content	Add content to the selected Letter Instance. The selected Letter Instance will contain Data Resource values that can be overwritten by a User.
Select Preview DRAFT Letter Instance	Select draft Letter Instance for preview.
Render PDF(s) with DRAFT watermark	Generate PDF of draft Letter Instance with watermark.
PDF Generation Successful?	Decision Point: Check if PDF generation was successful.
If No Notify user of errors	User is notified of errors prohibiting PDF rendering.
If Yes Preview DRAFT Letter Instance	Preview the PDF of draft Letter Instance.
Does Letter PDF look good?	Decision Point: Does the PDF preview meet requirements?
If No Edit Letter Instance Content	Return to editing the Letter Instance.
If Yes Mark Letter Instance as READY FOR REVIEW	Mark Letter Instance as READY FOR REVIEW.
Letter Instance Draft Complete Event	System records completion event; Letter Instance moves to READY FOR REVIEW state.
Template Version Updated?	Decision Point: If Template Version has changed, notify user.
Notify user New Template Version is AVAILABLE	User is warned of potential data loss if new Template Version is selected.
Use New Template Version?	Decision Point: Choose to update Letter Instance with new template version.
If Yes Letter Instance Updated with New AVAILABLE Template Version	Update Letter Instance with new template version.
If No User clicks to Cancel the Letter Instance	The Letter Instance is cancelled by the user and the process ends.

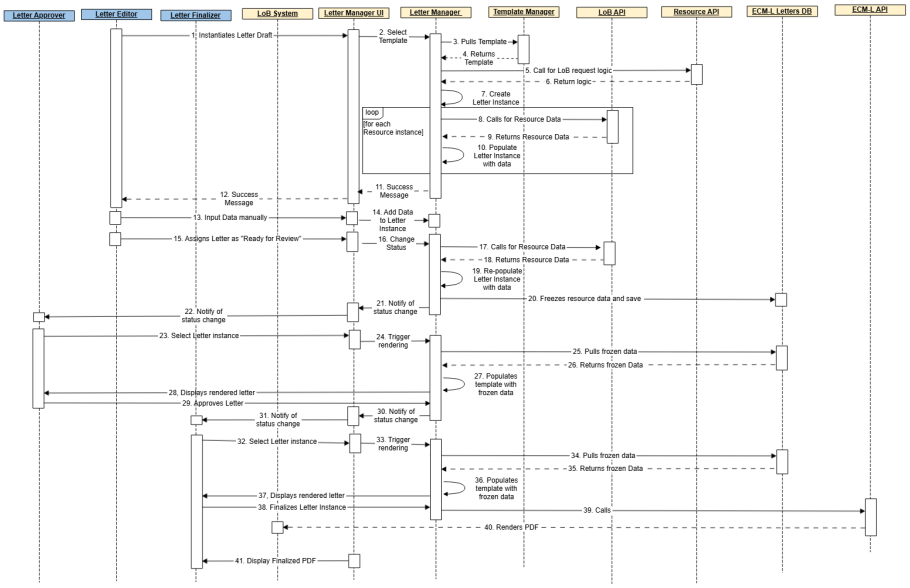
Letter Finalizer

Task	Description
User logs in to Letter Manager	Letter Finalizer logs into the Letter Manager to finalize letter instances.
View READY FOR REVIEW Letter Instance(s)	Display all letter instances that are in READY FOR REVIEW state.
Render PDF(s) with READY FOR REVIEW watermark	Generate PDF(s) with watermark for final review.
Finalize Letter Instance(s)?	Decision Point: Choose to finalize or reject the letter instance(s).
If Yes Mark Letter Instance(s) FINALIZED	Mark letter instance(s) as finalized.
Letter Instance Finalized Event	System records the event; letter instance moves to FINALIZED state.
If No Mark Letter Instance(s) REJECTED	Mark letter instance(s) as rejected.
Letter Instance Draft Rejected Event	System records the rejection; letter instance moves back to DRAFT state.

Process Sequence - Systems Event-Trace Description (SV-10c)

The following diagrams illustrate chronological sequences of events and interactions within different workflows of ECM-L, providing a dynamic view of system behavior over time.

Letter Creation Sequence (SV-10c)



Implementation View

The Implementation View (sometimes called the Development View) will be further completed alongside the Development of ECM-L.

Deployment View

Purpose	This section is to document any deviations or unique conditions that are not outlined by the standard practices of the BIP Platform which may be found in the Deployment View section of BIP Architecture Documentation (2001AM).
Acceptance Criteria	• Environments • System Configuration • System Monitoring and Metrics • System Logging and Auditing • Security Strategy
Diagram Type	UML Component or Block if needed

Environment Mapping

Environment	Description
DEV	Development Environment
TEST	Development testing environment
INT	Development Integration testing environment
IV&V/SQA	Independent Verification & Validation / System Quality Assurance
UAT	User Acceptance Testing
PAT	Pre-Acceptance Testing

PROD Development Test (PDT)	Environment used by Development to Test changes and patches
DEMO	Demonstrations
PREPROD	Environment used to test new releases prior to deploying to PROD
PERF	Performance testing
Cost of Living Adjustment (COLA)	Environment used to assist in determining Cost of Living changes
PROD Test (PRODTEST)	Environment to test PROD changes and patches prior to deploying to PROD
PROD	Production

System Configuration

No deltas from BIP standard practices.

System Monitoring and Metrics

No deltas from BIP standard practices.

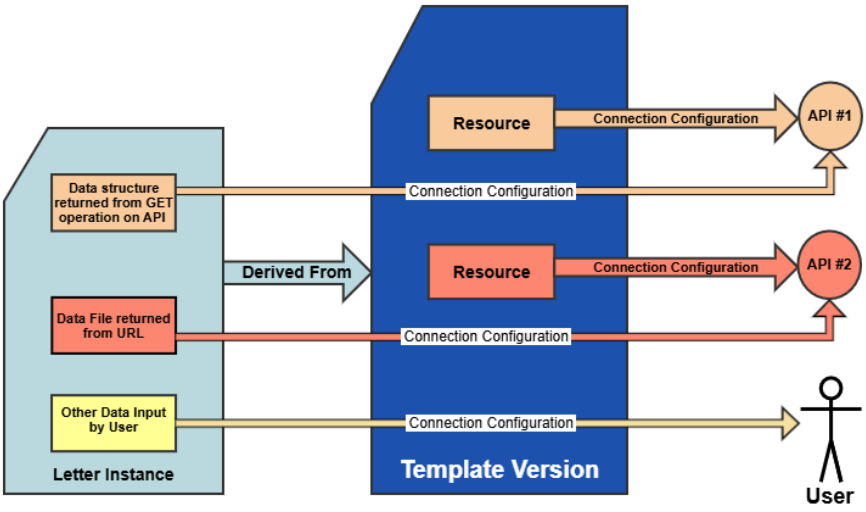
System Logging and Auditing

No deltas from BIP standard practices.

Long-Term Provenance

Letter Instance Provenance refers to the ability to chronicle how a Letter Instance was created, i.e., by whom, and when, while also tracing the origins of every element, including configured content as well as all data values input. A Letter Instance is derived from a Template Version, thus maintaining a comprehensive record of the Template Version, from which it was derived, including the Template Version's unique content and the Resource Version(s) it used, will ensure traceability of all the configured content. Chronicling the data input ensures the verifiable origin of the data necessary for auditing and validation purposes. For all data fields, the record will log the data/time of input, the source of data (e.g., the Resource Version from which the data originated (API or URL) or user, if manually input), and the actual data values entered in the Letter Instance.

The diagram below illustrates how each data element depicted in the Letter Instance (Orange, Red, Yellow) can be traced to its origin. For example, the Data Structure input in the Letter Instance can be traced to the Orange Resource (configured in the Template Version) which used a Connection Configuration to API #1 to access the data used to populate the Letter Instance. The Data File can be traced to the Red Resource (configured in the Template Version) which accessed the Data File from API #2 via the Connection Configuration and Other Data can be traced to the User who manually input the data in the Letter Instance.



Security Strategy

ECM-L is heavily dependent on policies concerning authentication and authorization and will utilize the **B**enefits **A**pplication **A**uthori**Z**ation framework (BAAZ) as the propagation mechanism for an LOB's authorization policies. BAAZ is a policy framework that uses Open Policy Agent (OPA) with a central policy store where LOBs can manage their own policies and any application can create its own policies and apply policies that are managed by a central security authority and appropriately audited. (See Benefit Applications Authorization Services). As an enterprise system, LOBs will have self-service capabilities for maintaining their own governance and security policies. An important part of the Intake Process will be for each LOB to establish a set of policies for that LOB, including policies for access to Applications and Workflows.

Recommendation

Analysis of Alternatives

Based on comprehensive analysis across technical, business, and operational dimensions, comparing the ECM-L solution against a solution making use of the Salesforce Platform, the recommendation is to proceed with this custom development approach to best meet the business's needs. This recommendation is supported by the following key findings:

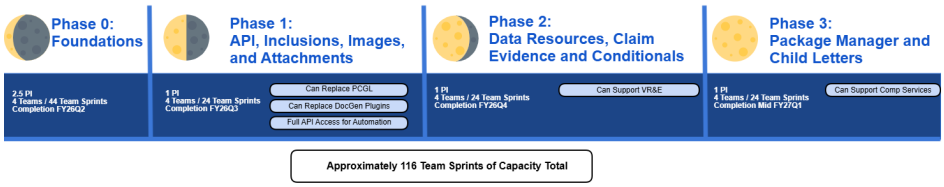
1. **Superior Technical Fit**: Provides significantly better performance at the required scale (approximately 40,000 users across 20 Lines of Business generating 20 million letters annually) and enables the strict multi-tenant isolation required between the different Lines of Business.
2. **Lower Total Cost of Ownership**: Provides a 22% cost advantage over Salesforce ($28.7M), primarily due to lower annual licensing and infrastructure costs. (see Costing Considerations.)
3. **Better Risk Profile**: Presents lower risks in areas critical to success, including performance at scale, future cost escalation, and vendor lock-in.
4. **Greater Long-term Flexibility**: Provides superior flexibility for future enhancements and integrations without platform constraints.
5. **Stronger Alignment with Enterprise Strategy**: Better supports the organization's strategic priorities for control over core systems, vendor diversification, and cost optimization.

Custom development offers an unparalleled combination of performance optimization, multi-tenant isolation, and customization flexibility, allowing us to achieve all of our goals for less.

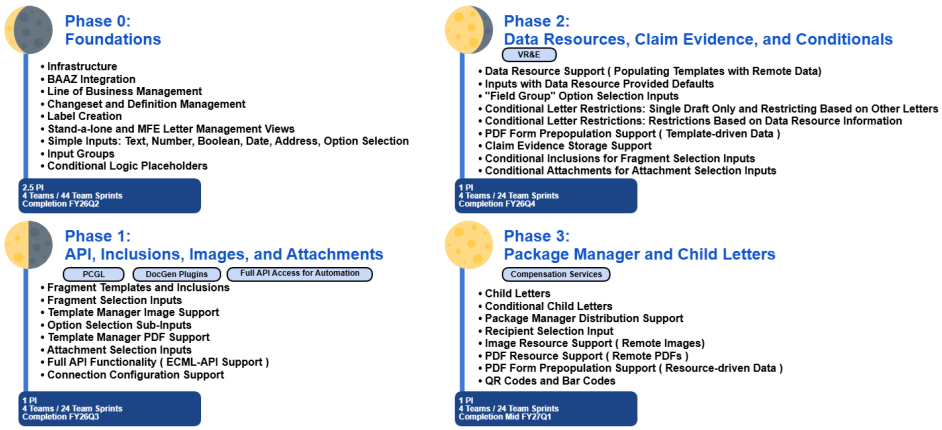
Recommendation

Phases and Roadmap

Summary of Phases



Functionality Breakdown



Phase Map

Functionality Group	Functionality	Jira Ticket
Infrastructure	New Tenant Apps, Repos, etc.	VBMSR-32947 Capability to Establish Infrastructure to support ECML
BAAZ	BAAZ Integration Intake Process	VBMSR-32948 Capability to Establish Infrastructure to support ECML
Line of Business Management	Create Workflow	VBMSR-33097 Capability to Manage Workflow
	Create Key Parameter Type	VBMSR-33098 Capability to Manage Key Parameter Types
Changeset and Template Management	Create Changeset	VBMSR-32955 Capability to Create a Changeset and Update Its Attributes VBMSR-32959 Capability to Cancel and Recall a Changeset VBMSR-33154 Capability to Self-Assign a Changeset
	Create, Update, Retire Template	VBMSR-32956 Capability to Manage Template Versions in a Changeset VBMSR-32961 Capability to Add, Update Simple Attributes on a Template VBMSR-32963 Capability to Retire a Template Version
	Create and Run Test Scenarios	VBMSR-32957 Capability to Test Template Versions in a Changeset
	Schedule Changeset	VBMSR-32958 Capability to Validate and Schedule a Changeset
	Create, Align Template Label	VBMSR-33769 Ability to Update Attributes of a Template
	Get Template List	VBMSR-33255 Capability to View a List of Templates for a Line of Business VBMSR-33754 Capability to View Templates that are associated to a Fragment Template Version
	Label Creation	VBMSR-33099 Capability to Manage Template Labels
Stand-alone and MFE Letter Management	Create, Edit, Delete Letter	VBMSR-32964 Capability to Access Stand Alone Letter Manager Portal VBMSR-32966 Capability to Process a Letter Instance Through Its Lifecycle
	Render Letter	VBMSR-33100 Capability to View a PDF Rendered by ECM-L API
	Finalize Letter	VBMSR-32966 Capability to Process a Letter Instance Through Its Lifecycle
Input Support	Conditional Logic	VBMSR-33105 Capability to Support Conditional Content in a Template Version
Simple Input Support	Text Inputs	VBMSR-33101 Capability to Support Simple Input Fields on a Template Version VBMSR-32967 Capability to Manage Simple Inputs on a Letter Instance
	Number Inputs	VBMSR-33101 Capability to Support Simple Input Fields on a Template Version VBMSR-32967 Capability to Manage Simple Inputs on a Letter Instance
	Boolean Inputs	VBMSR-33101 Capability to Support Simple Input Fields on a Template Version VBMSR-32967 Capability to Manage Simple Inputs on a Letter Instance
	Date/Time Inputs	VBMSR-33101 Capability to Support Simple Input Fields on a Template Version VBMSR-32967 Capability to Manage Simple Inputs on a Letter Instance
	Address Inputs	VBMSR-33410 Capability to Support Address Inputs on a Template VBMSR-33256 Capability to Enter Values for Input Groups, Addresses and Option Selection Inputs on the Letter Interview Screen
	Phone Inputs	VBMSR-33381 Capability to Support Phone Input Fields on a Template Version VBMSR-33414 Capability to Enter Phone Inputs on the Letter Interview Screen
	Option Selection Inputs	VBMSR-33104 Capability to support Simple Option Selection Inputs on a Template VBMSR-33256 Capability to Enter Values for Input Groups, Addresses and Option Selection Inputs on the Letter Interview Screen

Complex Input Support	Input Groups	VBMSR-33409 Capability to Support Input Groups on a Template Version VBMSR-33256 Capability to Enter Values for Input Groups, Addresses and Option Selection Inputs on the Letter Interview Screen
	Attachment Selection Inputs	VBMSR-33645 Capability to Configure Attachment Selection Inputs on a Template Version VBMSR-33646 Capability to Populate an Attachment on a Letter Instance VBMSR-33701 Capability to Configure Conditional Rules in Attachment Selection Inputs
	"Field Group" Option Selection Inputs	... need the initiative for template manager ... VBMSR-33716 Capability to Interact with the List Builder Input Type in the Letter Manager
	Option Selection Sub-Inputs	
	Recipient Selection Input	/VBMSR-33717 Capability to Add and Configure the Recipient Input Type in Template Manager VBMSR-33718 Capability to Interact with the Recipient Selection Input Type in Letter Manager
	Inputs with Data Resource Defaults	VBMSR-34383 -Capability to Assign Data Resources as Default Values for Input Fields
Inclusions	Template Inclusion	VBMSR-33586 Capability to Manage Fragment Template on a Template VBMSR-33754 Capability to Templates that are associated to a Fragment Template
	Fragment Selection	VBMSR-33630 Capability to Allow a User to Manage Fragment Selection Inputs within the Content of a Template VBMSR-33629 Capability to Allow a User to Select a Fragment Selection Input to be Included in a Letter Instance VBMSR-34328 Capability to Override the Conditional Rules Assigned to Fragment Selection Inputs within Fragments
	Conditional Inclusions	
Image Resource Support	Upload Image Support	VBMSR-33758 Capability to Manage Images in a Template Version VBMSR-33644 Capability to Support File Versions on a Template
Integration Support	Full API Functionality	VBMSR-33385 Capability to Process Letter Instances by ECM-L API VBMSR-33100 Capability to View a PDF Rendered by ECM-L API
Connection Configuration Support	Connection Configuration Support	VBMSR-32951 Capability to Manage Connection Configurations
Attachment Support	File Version or "DGMT-Style Attachments"	VBMSR-33644 Capability to Support File Versions on a Template
	Conditional Attachments	VBMSR-33701 Capability to Configure Conditional Rules in Attachment Selection Inputs VBMSR-33702 Capability to Apply Conditional Rules for Attachments in Letter Manager
	Attachment Field Population	... need template manager requirement ...
	Remote Image Support	
	Resource-Derived Attachment Form Population	
Restrictions	Letter Restrictions for Single and Existing Letters	VBMSR-34109 Capability to Restrict Letter Create based on Rules
	Conditional Restrictions	VBMSR-34109 Capability to Restrict Letter Create based on Rules VBMSR-33719 Capability to Restrict Manual Creation and/or Finalization of Letter Instances

Populate Template with Remote Data	Text Fields (Data Resource)	VBMSR-34421 Capability to Manage Data Resources in the Resource Manager VBMSR-34403 Capability to Configure Data Resource Placeholders on a Template Manager
	Number Fields (Data Resource)	VBMSR-34421 Capability to Manage Data Resources in the Resource Manager VBMSR-34403 Capability to Configure Data Resource Placeholders on a Template Manager
	Boolean Fields (Data Resource)	VBMSR-34421 Capability to Manage Data Resources in the Resource Manager VBMSR-34403 Capability to Configure Data Resource Placeholders on a Template Manager
	Date Fields (Data Resource)	VBMSR-34421 Capability to Manage Data Resources in the Resource Manager VBMSR-34403 Capability to Configure Data Resource Placeholders on a Template Manager
	Physical Address Fields (Data Resource)	VBMSR-34421 Capability to Manage Data Resources in the Resource Manager VBMSR-34403 Capability to Configure Data Resource Placeholders on a Template Manager
	Field Groups (Data Resource)	VBMSR-34421 Capability to Manage Data Resources in the Resource Manager VBMSR-34403 Capability to Configure Data Resource Placeholders on a Template Manager
Child Letters	Child Letters	
	Conditional Child Letters	
Bar Code Support	QR Codes	VBMSR-34252 Capability to Manage Unique Scannable Identifier (USI) on Template Version
	Bar Codes	VBMSR-34252 Capability to Manage Unique Scannable Identifier (USI) on Template Version
Storage and Distribution	Claim Evidence Support	VBMSR-34108 Capability to Automatically Upload Finalized Letters to Claims Evidence
	Package Manager Support	VBMSR-33717Capability to Add and Configure the Recipient Input Type in Template Manager VBMSR-33718 Capability to Interact with the Recipient Selection Input Type in Letter Manager