Partner Licensing Platform Developer Guide (Developer Preview)

Version 60.0, Spring '24

@salesforcedocs

Last updated: January 12, 2024

CONTENTS

CHAPTER 1 Partner Licensing Platform (Developer Preview)

In this chapter ...

The Partner Licensing Platform (PLP) is a robust new platform that revolutionizes how Salesforce Partners and ISVs price, license, and distribute their applications. Product owners, architects, and developers who build apps for AppExchange can transform their business and products with the power of PLP.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Our forward-looking statement applies to partner business org provisioning. Make your purchasing decisions based on currently available technology.

Now for the first time, you can develop and distribute multiple license types for your package that entitle specific features, without breaking up your package structure. With this functionality you can increase and diversify your revenue stream by selling multiple tiers, versions, or supplements for your products via these license types.

You can also create licenses that can be assigned only to specific types of users. For example, a license for internal Salesforce users with complete functionality, and a separate license for external Experience Cloud users with the same or limited functionality, for a lesser price. You can potentially recover revenue for products that were previously only contractually restricted.

This onboarding guide outlines how to sign up for the developer preview and design and implement your new licensing strategy. Starting now, you can get ready to take advantage of the incredible capabilities of the next generation of licensing for Salesforce Partners.

What's Available in the Developer Preview?

The following features are available in the Summer '22 release:

- Custom foundation and supplement permission set licenses
- License expiration policies
- Licensed custom permissions
- User license restrictions

Partner business org provisioning of custom permission set license seats is an expected eventual capability of the Partner Licensing Platform.

For info and updates about the developer preview, join the Partner Licensing Platform Developer Preview Partner Community group. For general news about PLP, join the Partner Licensing Platform Partner Community group.

SEE ALSO:

TrailheaDX '21 Presentation: Next Generation Licensing for ISV Partners

Salesforce Video: Partner Licensing Platform Demo

Enable the Partner Licensing Platform (Developer Preview)

Review requirements for the developer preview and set up necessary orgs before submitting a participation request.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Scope of the Developer Preview

The developer preview isn't available for your production development and packaging orgs. To participate in the developer preview, you must create **new** orgs for using and testing the new Partner Licensing Platform.

You're encouraged to use your existing package code to see how you would transform your package and product with the new licensing capabilities. You can also consider creating a new repository or branch in your package code source control for implementing the new functionality alongside your production package development.

Alternatively, we prepared a demo application that you can use to test out the new functionality with your new developer preview orgs.

Requirements

The Partner Licensing Platform supports both 1GP and 2GP. To participate in the developer preview, you must have access to an Environment Hub (Env Hub) and a Developer Hub (Dev Hub). Both hubs are available to all AppExchange partners as part of your Partner Business Org (PBO).

Participants must be familiar with scratch org-based development using the Salesforce Developer Experience command-line interface (SFDX CLI). Scratch orgs are required as development and testing environments during the developer preview.

Create Your New Orgs

You must create the following orgs for end-to-end development, packaging, and testing with the Partner Licensing Platform enabled:

- A new Partner Enterprise org with Dev Hub enabled
  - See Create an Org from the Environment Hub and create a Test/Demo org using the Partner Enterprise edition.
  - See Enable Dev Hub Features in Your Org to enable your Dev Hub.
  - See Authorization to authenticate the Salesforce CLI to your Dev Hub.
- A new Partner Developer org with a namespace prefix that starts with **plpdp_**
  - See Create an Org from the Environment Hub and create a Development org using the Partner Developer edition.
  - See Register a Namespace and make sure that the namespace prefix you choose for your Partner Developer org starts with plpdp_ (such as "plpdp_mytest").
  - If testing with 1GP, use this Partner Developer org as your packaging org.
  - If testing with 2GP, link a namespace to your Dev Hub to allow creation of second-generation packages with your new namespace.

Sign Up for the Developer Preview

After you create the required orgs, make note of their org IDs. Then, submit a participation request via the Partner Licensing Platform Developer Preview Partner Community group. After the request is reviewed and approved, Salesforce will enable the developer preview in the new orgs that you created.

Note that there is limited space and Salesforce may close the developer preview to new partners when the current cohort is full.

Set Up the Development Environment

In the developer preview, you'll create your new custom permission set licenses in scratch orgs created from your new Dev Hub. After your request to join the developer preview is approved, the Partner Licensing Platform will be enabled for your Dev Hub. This will allow you to create scratch orgs with the `PartnerLicensingPlatform` feature enabled in the scratch org definition file. For more information about enabling features in scratch orgs, see Build Your Own Scratch Org Definition File, or take a look at the demo application scratch org definition.

After you implement your new licenses and settings, you're ready to create a new package with this metadata. For 1GP, deploy the metadata to the new packaging org and create a new package. For 2GP, add the metadata source to your package directory's source and create a new package using SFDX.

Finally, when you're ready to install and test your new package from the package subscriber's perspective, use a new scratch org created from your new Dev Hub. Packages with the new functionality can only be installed in scratch orgs that have the `PartnerLicensingPlatform` feature enabled in their scratch org definition files.

Note: To use the Partner Licensing Platform, you must use API version 55.0 or later.

Quick Start: Get Started with the Partner Licensing Platform (Developer Preview)

Get up and running with the Partner Licensing Platform with a demo application.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Our forward-looking statement applies to partner business org provisioning. Make your purchasing decisions based on currently available technology.

After your development environment is set up, you can use the demo application to try out the Partner Licensing Platform (PLP).

The demo app contains all of the code and metadata components needed to demonstrate how an AppExchange application's functionality can be gated using licensed custom permissions and custom permission set licenses. We go into detail about these new components in Partner Licensing Platform Components and Concepts (Developer Preview), but you can set up and use the demo app immediately.

Detailed instructions for setting up and using the demo application are located in the README file of the demo application repository. Follow the steps outlined there to continue.

Considerations During the Developer Preview

During the developer preview, custom permission set licenses can only be tested via an installed package, as a subscriber of your package. For testing in the developer preview, the subscriber org where the managed package is installed is provisioned a default count of 10 seats for each custom permission set license in the package. When the functionality is available, you'll be able to provision a number of seats for your custom permission set licenses to each of your subscribers from your partner business org.

Using the Demo App

After you've finished setting up the demo app, you're ready to see how the Partner Licensing Platform works for a subscriber. First, log in to your subscriber org using the admin user and assign the Feature Access Demo permission set to the standard user. Then, log in to your subscriber org using the standard user and navigate to the Feature Access Visualforce page via the App Launcher in Lightning Experience.

The Feature Access page simulates access to package features that are gated behind licenses. Clicking the buttons allows you to see which features are accessible to the current user. Without the proper custom permission set licenses and permission sets assigned to the user, the user can't access the features on the page.

Now log in to your subscriber org as the admin user. Assign the **Feature_A** custom permission set license to your standard user. Now, that user is entitled to use Feature A in that org.

While the standard user is **entitled** to Feature A via a license assignment, the user still can't access Feature A until the admin further **grants access** to the user to Feature A via the **FeatureA User** permission set.

This grant is only possible after the admin has assigned the **Feature_A** custom permission set license to the standard user. The custom permission set license **entitles** the standard user to Feature A, and then the permission set **grants access** to the standard user for Feature A. This process of assigning licenses and permission sets results in a two-step access check for accessing a licensed feature.

Now, you'll see that the standard user can access Feature A, but not Feature B. To further illustrate how licenses and permission sets work together, try to grant access to the standard user to Feature B by assigning the **FeatureB User** permission set. This assignment isn't allowed because access to Feature B can only be **granted** via permission set to users who are first **entitled** to use Feature B via a license assignment. In this way, users are only able to **access** features that they're both **entitled** and **granted**.

You can explore combinations of custom permission set license and permission set assignments with the standard user and see how their access changes for gated features. Later in this guide, the concept of entitlements and granting access is explored in more detail.

In this demo app, we demonstrated how to gate access to your package features using custom permission set licenses and licensed custom permissions. You can also use this demo later as reference when you implement your package licensing.

What's Next?

Now that you've used the demo app to try out PLP, it's time for you to take advantage of the power of the platform to transform your own business and products. You can develop and distribute multiple license types for your package that entitle specific features, without breaking up your package structure. With this functionality, you can increase and diversify your revenue stream by selling multiple tiers, versions, or supplements for your products via these license types. You can even create separate licenses that can be assigned only to specific types of users, such as internal Salesforce users and external Experience Cloud users, separately.

The following sections dive into the data model of the new Partner Licensing Platform, covering how the platform works and how this new metadata and permissioning lifecycle is structured. Then, we walk you through the end-to-end process for how to redesign your product licensing with these capabilities using an example product.

As you go through the journey, you're welcome to ask questions and collaborate on the Partner Licensing Platform Developer Preview Partner Community group. For general news about PLP, join the Partner Licensing Platform Partner Community group.

Partner Licensing Platform Components and Concepts (Developer Preview)

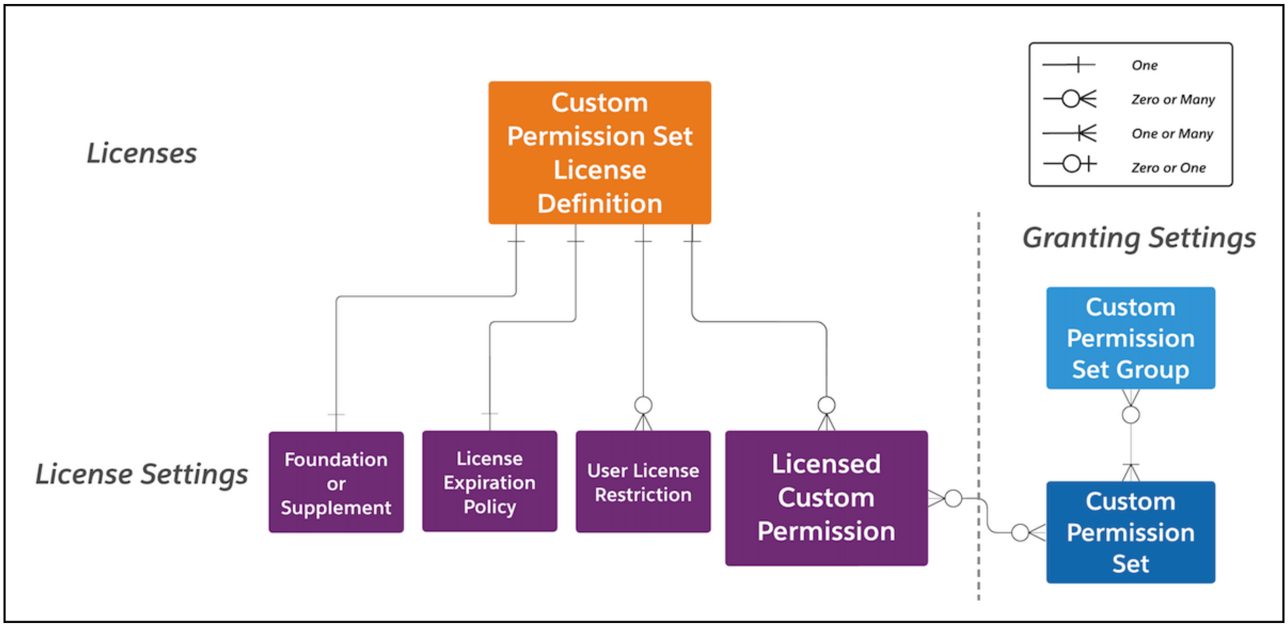
Learn about the components of the Partner Licensing Platform and how they're connected.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Our forward-looking statement applies to partner business org provisioning. Make your purchasing decisions based on currently available technology.

Before you dig into this material on the Partner Licensing Platform, we recommend that you take the Salesforce Licensing Trailhead module for an overview of licensing at Salesforce.

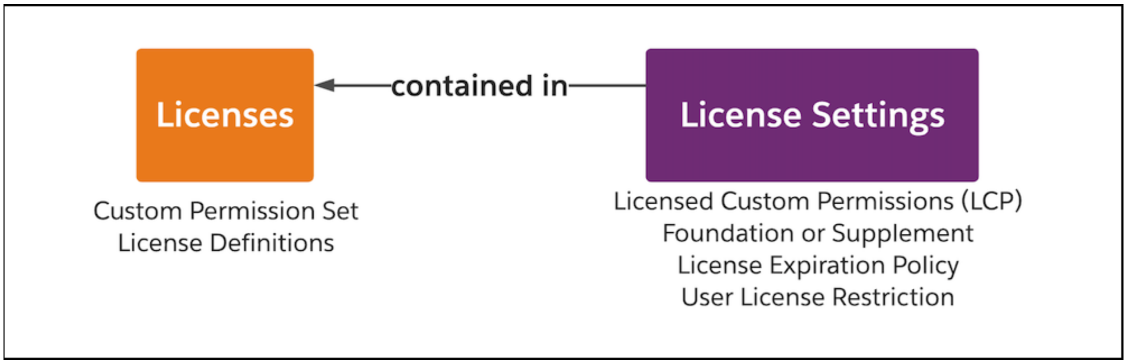
This diagram summarizes the main components that you create and how they're structured.



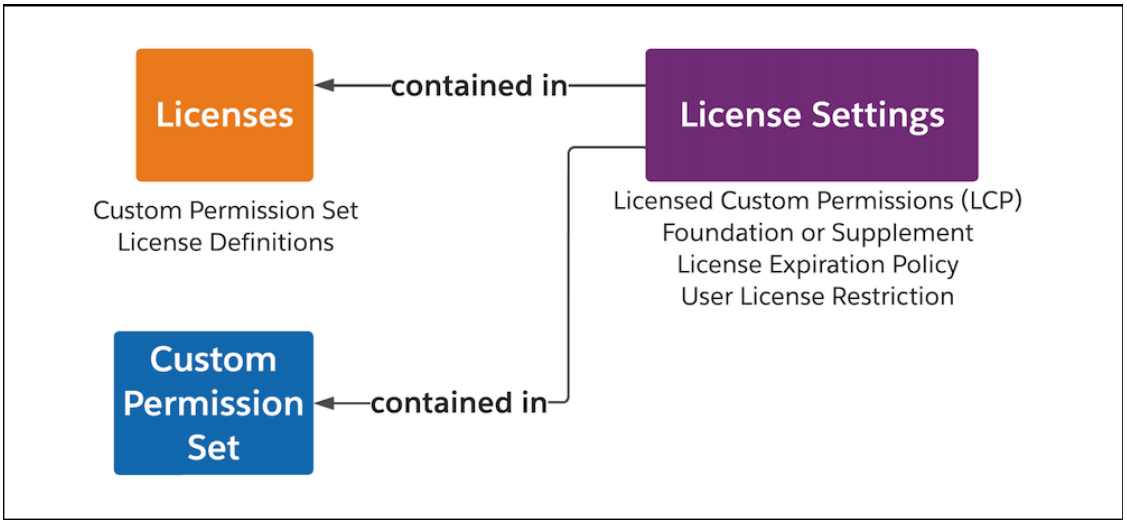
At a high level, you create custom permission set license definitions that are a part of your managed package metadata, just like custom objects and permissions. Then, you create license settings, such as licensed custom permissions, that are contained in the custom permission set license definitions. The licensed custom permissions are also contained within custom permission sets, which in turn can be bundled into permission set groups. When creating a custom permission set license definition, you also specify whether it's a foundation or supplement license and which user licenses can be assigned the license. We go into more detail on these settings in the other topics in this section.

Let's go one step further and look in this diagram how these components and their relationships are used to gate a feature in your package.

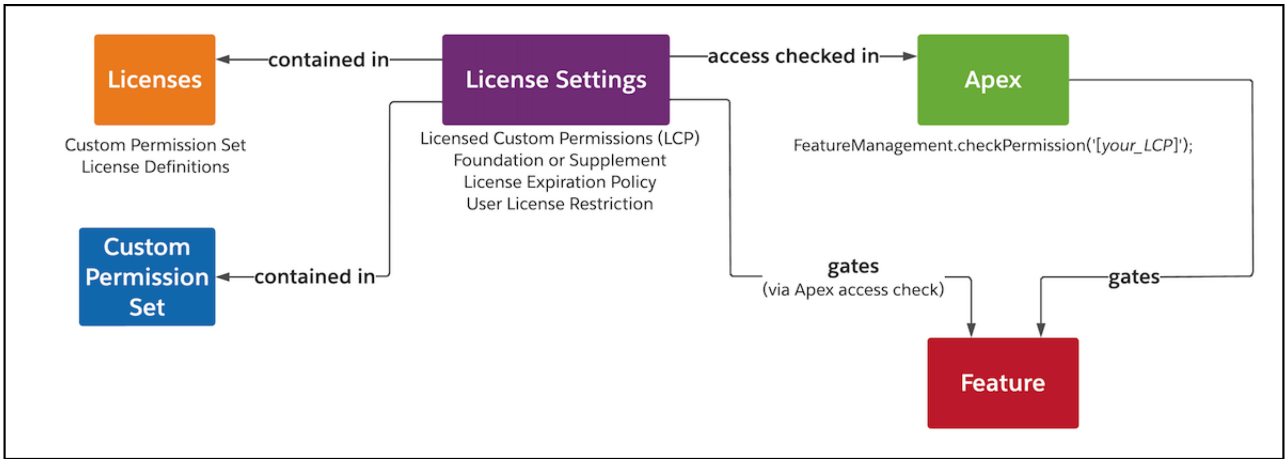
For example, you have a feature that you want to sell and provision. First, you create a licensed custom permission in your development scratch org, then put this custom permission into a new custom permission set license definition, which **entitles** this custom permission.



Next, you put the custom permission into a custom permission set, which **grants** this custom permission.



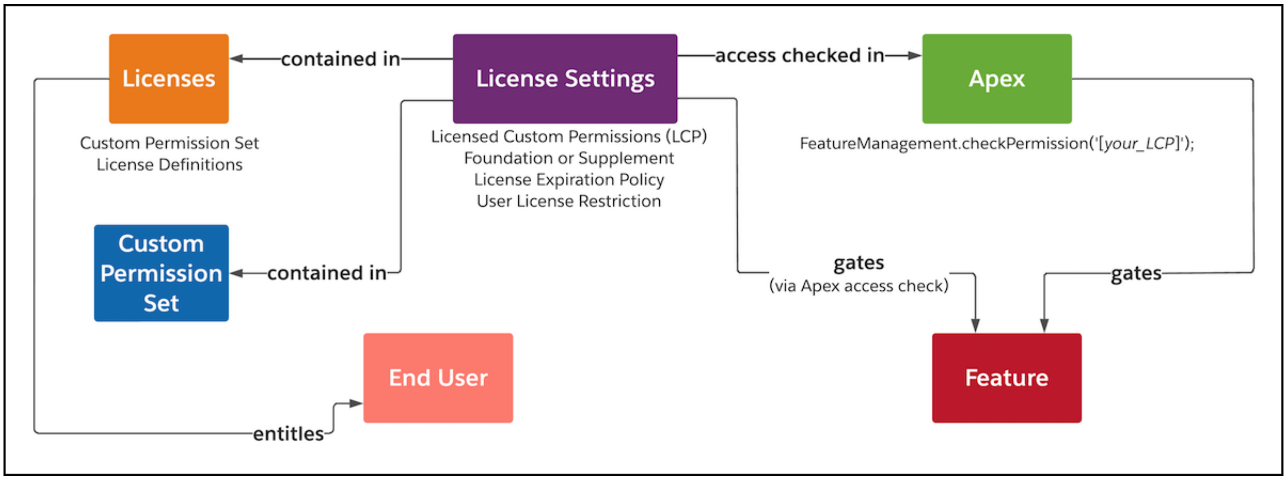
Finally, you put an access check for the licensed custom permission in your Apex code for a specific feature. This access check gates access to the feature.



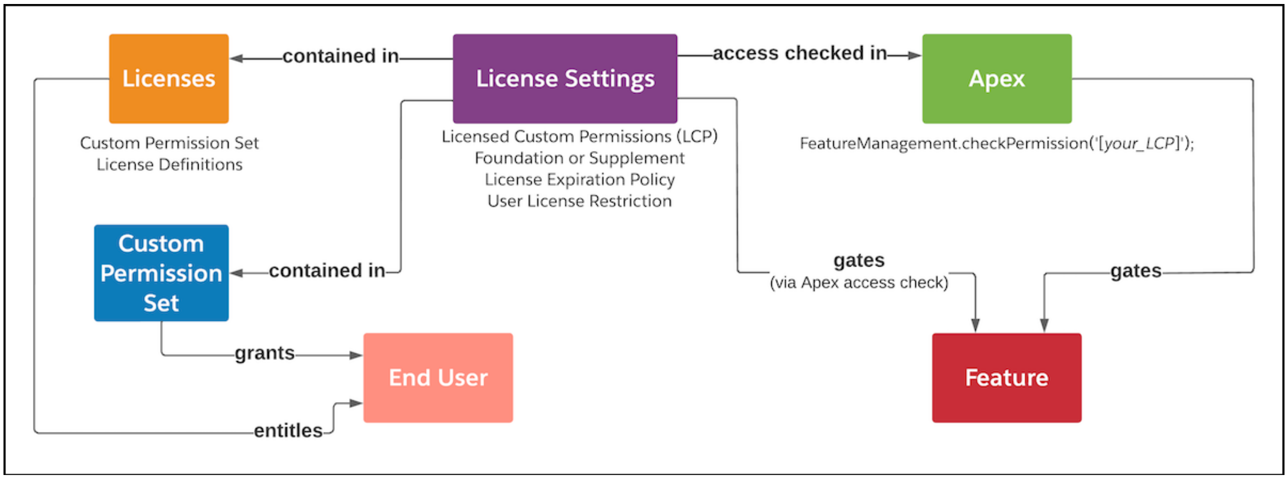
All that's left is uploading and shipping your managed package with these entities. The configuration of licensing as package metadata is separate from provisioning specific quantities of seats for your licenses for a subscriber.

Next, when the functionality is available, you'll provision a number of seats for your custom permission set licenses to each of your subscribers from your partner business org.

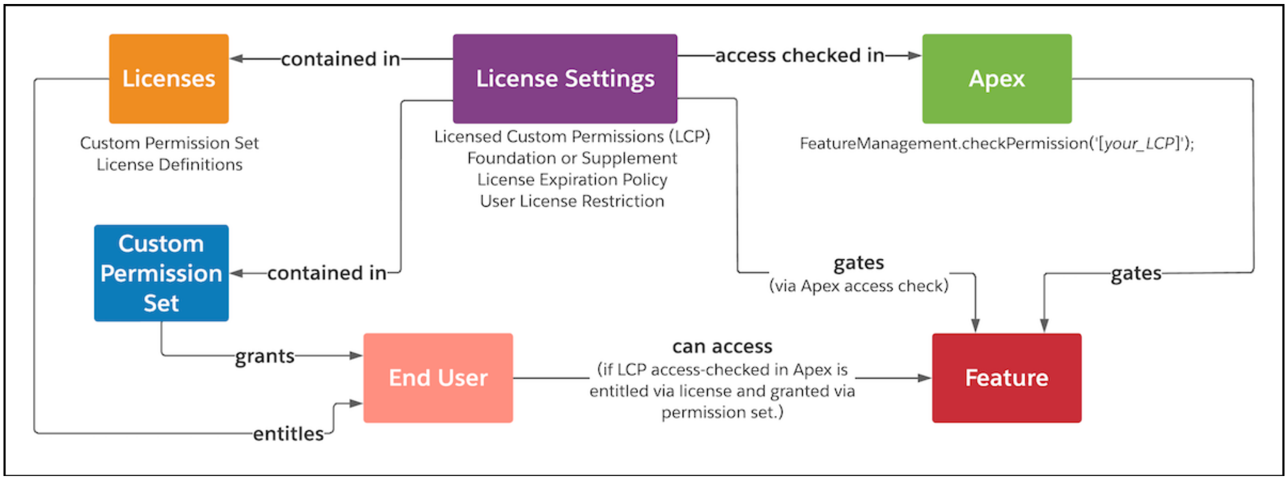
On the subscriber's side, they receive the new metadata as part of your package and a number of custom permission set license seats. First, the subscriber admin assigns the custom permission set license to an end user, which **entitles** the user the licensed custom permission, or feature. This entitlement allows the user to be granted the permission by the admin in the next step.



Next, the admin assigns the custom permission set to the end user, which **grants** that user the licensed custom permission, or the feature. Since this permission set contains a licensed custom permission, it can only be assigned to the user after the user has a license assigned with the licensed custom permission.



The end user can now access the feature gated by the licensed custom permission, because the user is both **entitled** by a license assignment and **granted** access by a permission set assignment.



This process of assigning licenses and permission sets results in a two-step access check for accessing a specific feature. Since the quantity of each license available to a subscriber is provisioned from your partner business org, once available, this mechanism allows both you and your subscribers granular control for which features are purchased and then how they are entitled and granted to each user.

Note: Partner business org provisioning of custom permission set license seats isn't available in the developer preview. The channels and interface for provisioning custom permission set license seats are to be determined and will likely be separate from the current License Management App. For testing in the developer preview, the subscriber org where the managed package is installed is provisioned a default count of 10 seats for each custom permission set license in the package.

In these topics, we go into more detail about individual licensing components as well as entitlements and grants.

Licensing Components (Developer Preview)

Review the different components that are created as part of your licensing and packaging structure and how these pieces fit together.

Entitlements and Grants (Developer Preview)

For users to be able to use a package feature, they must first have access to the overall package. Then, they must be entitled to the feature with a license assignment, and then granted access to the feature with a permission set assignment.

Licensing Components (Developer Preview)

Review the different components that are created as part of your licensing and packaging structure and how these pieces fit together.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Our forward-looking statement applies to custom supplement permission set licenses, user license restrictions, and licensing and provisioning integration with Sales Cloud. Make your purchasing decisions based on currently available technology.

Custom Permission Set Licenses

A custom permission set license lets Salesforce Partners and ISVs incrementally sell user-level functionality for individual users. By assigning custom permission set licenses to users, admins can extend a user's entitlements, just like Salesforce-managed permission set licenses.

Admins in subscriber orgs can view their custom permission set licenses in the Company Information page in Setup, after their Salesforce org is provisioned custom permission set licenses. Then, they can assign custom permission set licenses, along with their related custom permission sets, to users who need the permissions in the license.

When an admin assigns a user a permission set that grants access to package components, the system first validates whether the user has a custom foundation permission set license that allows access to the package namespace. For the original managed package licenses, package access isn't validated until the user attempts to access the components. This delay can cause an inconsistent authorization experience, because admins can assign users permission sets that grant access to components in a package, while those users may not have access to the package itself via a license. For more information on package access validation when migrating to the Partner Licensing Platform, see this topic.

User-Level License Settings

User-level license settings gate individual features of your product for each user and are **entitled** by a license.

Licensed Custom Permission

A regular custom permission but with the **License Required** checkbox enabled. The field indicates that this custom permission is licensed and must first be entitled via a custom permission set license for a user to be able to access it via a custom permission set.

Foundation or Supplement

You can indicate whether the custom permission set license is a foundation or supplement license. The default, custom foundation permission set licenses, are like managed package licenses. They entitle and grant access to a user to enter the package namespace, which is required before any further use of the package or its components. A user's managed package license assignment can be replaced with a custom foundation permission set license and the user will retain the same level of access to all the "foundation" features of the package. Foundation features are the package components other than licensed custom permissions, such as non-licensed custom permissions, custom objects, Apex classes, and Visualforce pages. Custom foundation permission set licenses can also entitle the user to specific licensed features. Note that a permission set assignment is still required to grant the user access to specific components of the managed package, such as custom objects.

Custom supplement permission set licenses can only entitle the user to specific licensed features and not the overall package. Supplement licenses are usually used for additional functionality that you want to sell on top of your foundation features and access to your package, as these licenses require the user to have at least one foundation license assigned first.

User License Restriction

You can specify the users that can be assigned the custom permission set license using user license restriction categories. For example, you can restrict the license to only be assigned to internal Salesforce users or external Experience Cloud users. For more information on the available categories and the included user licenses, see User License Restriction Categories (Developer Preview).

License Expiration Policy

You can choose whether to allow or block package access when all custom foundation permission set licenses expire. If you allow package access, users with the expired licenses can continue to access the package namespace and its components. However, the subscriber admin can't assign any more of these licenses to users and can't reassign the expired license to a user after it's removed. If you block package access, users with the expired licenses can no longer access the package namespace or its components upon expiration. This option is how the original managed package licenses operate.

Note: If only some of the licenses have expired, all licenses continue to allow access to the package namespace and its components until all licenses expire.

Permission Sets and Permission Set Groups

Permission set and permission set groups **grant access** to a license's settings, such as a licensed custom permission, or other regular components of your package.

Permissions Sets

Permission sets are groups of permissions and settings that extend users' functional access without changing their profiles. You create custom permission sets in your package that contain the licensed custom permissions. After a user becomes **entitled** to permissions via a license, admins use permission sets to actually **grant access** to these permissions. Permission sets are defined in the managed package as part of the package metadata.

Permission Set Groups

Permission set groups bundle permission sets together, for example, based on user job functions. Users assigned to the permission set group receive the combined permissions of all the permission sets in the group. Depending on the personas for whom you're creating your features, consider using permission set groups. Permission set groups provide admins with flexibility, since the admins can choose to mute specific permissions in a permission set group, depending on their business needs.

Order Management

You can use the Sales Cloud functionality that comes with your partner business org or another order management system (CRM). We recommend that you use Sales Cloud for its enhanced functionality and for built-in licensing and provisioning integration that may be available in a future release.

SEE ALSO:

Salesforce Help: Permission Sets

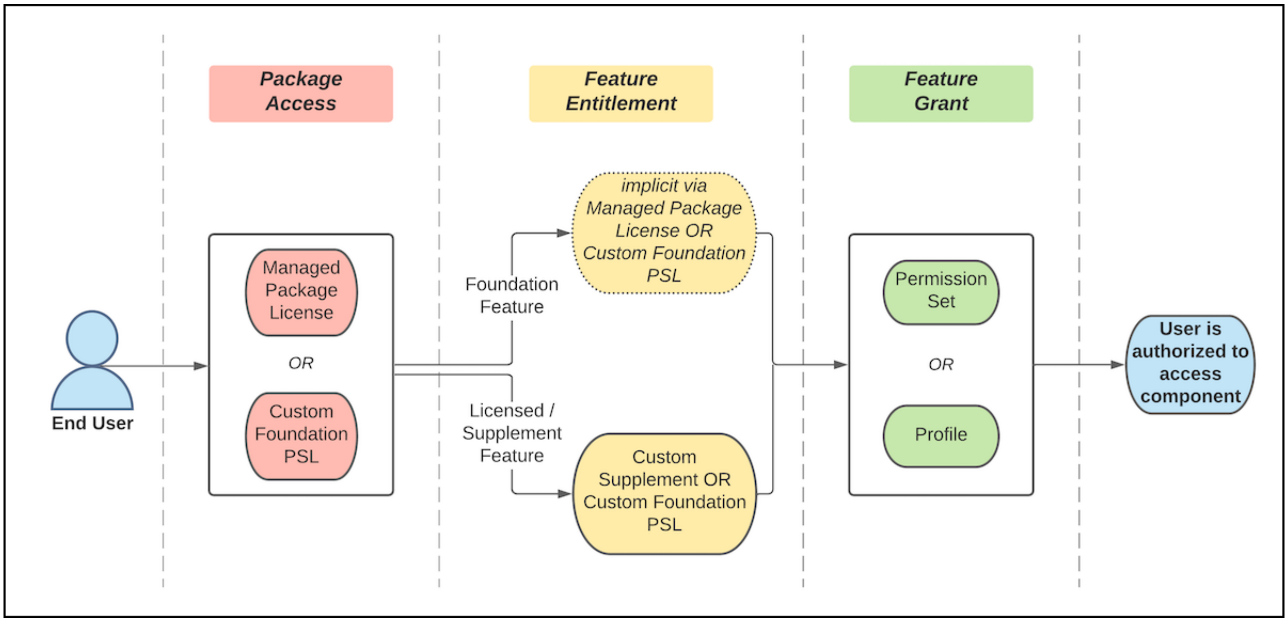
Salesforce Help: Permission Set Groups

Entitlements and Grants (Developer Preview)

For users to be able to use a package feature, they must first have access to the overall package. Then, they must be entitled to the feature with a license assignment, and then granted access to the feature with a permission set assignment.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Our forward-looking statement applies to partner business org provisioning. Make your purchasing decisions based on currently available technology.


Package Access
Feature Entitlement
Feature Grant
End User
Managed Package License
OR
Custom Foundation PSL
Foundation Feature
implicit via Managed Package License OR Custom Foundation PSL
Licensed / Supplement Feature
Custom Supplement OR Custom Foundation PSL
Permission Set
OR
Profile
User is authorized to access component

Package Access

Package access is the ability for a user to enter the namespace of a package. This access doesn't include further entitlements or grants necessary for the user to access any components of the package. However, it's necessary for a user to first have package access before receiving any further use of the package or its components.

Either a managed package license or a custom foundation permission set license (PSL) can provide package access to a user for a given package or namespace. A permission set or profile isn't necessary to give the user package access.

Foundation Feature Entitlement

Managed packages primarily contain foundation features, such as non-licensed custom permissions, custom objects, Apex classes, and Visualforce pages. All components of your package are considered foundation features, except for licensed custom permissions. Foundation features aren't explicitly licensed or named in a custom permission set license, which means a user doesn't require an explicit entitlement to access these features. Their entitlement is given to the user implicitly via the assignment of a managed package license or via a custom foundation permission set license. Both of these licenses provide implicit entitlement for all foundation features in your package, including as your package changes over time.

Licensed Feature Entitlement

Licensed or supplement features are features that are explicitly licensed. These features are gated with licensed custom permissions that are named in a custom permission set license. A user must have an explicit entitlement to access these features. When the admin assigns the user a custom permission set license, it provides the **entitlement** to the license settings or features it contains. Admins can then grant users access to the features they're entitled to with a permission set. The assigned licenses define the maximum set of features entitled to the user, but don't grant access to those features. The only package components that can explicitly be both entitled and granted for end users are licensed custom permissions.

Licensed features can be included in either a custom foundation or a custom supplement permission set license. Custom supplement permission set licenses don't give the user package access or entitlement to access the foundation features of the package. They can only give entitlement for licensed or supplement features of a package.

Feature Grant

Finally, the admin must assign the user a permission set to **grant access** to the license settings (or features) already entitled by a license assignment. Now, the user has access to the package and is authorized to access this feature.

This process of assigning licenses and permission sets results in a two-step access check for accessing a specific feature. Since the quantity of each license available to a subscriber is provisioned from your partner business org, once available, this mechanism allows both you and your subscribers granular control for which features are purchased and then how they are entitled and granted to each user.

For example, a subscriber purchases five seats of a custom permission set license that contains three features. The subscriber admin can assign one seat to a user, giving the user the **entitlement** for those three features. Then the admin can assign a permission set to the user, giving them the **grant** for only one of those features. In this scenario, the admin has authorized the user to access only one feature, and not the other two available with the license.

Design Your Package Licensing Structure (Developer Preview)

To make sure that your license development process proceeds as smoothly as possible, design the feature breakdown of your product and its associated licenses carefully before you begin implementation.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Outline Pricing Feature Strategy (Developer Preview)

Define what features and functionality you want to sell as part of your product offering. Then, map these features to the products that you'll sell.

Identify Licenses (Developer Preview)

After you develop your feature pricing strategy, identify the licenses that each product contains. Your licenses contain the underlying features that your products entitle.

Identify License Settings Use Cases (Developer Preview)

Identify the correct settings to include in your licenses that gate specific functionality so that users have the access to the intended features.

Map Settings to Licenses (Developer Preview)

After you confirm a list of license settings for your features, decide which settings to include in which custom permission set licenses.

Design Permission Sets (Developer Preview)

The design of permission sets is based on your intended end-user personas and the appropriate access they require to perform their tasks. The goal is to enable your customers to have a convenient administration experience.

Review Your Structure (Developer Preview)

Review your design to ensure that your licensing and permissions are structured as you need for your feature pricing and licensing strategy.

Outline Pricing Feature Strategy (Developer Preview)

Define what features and functionality you want to sell as part of your product offering. Then, map these features to the products that you'll sell.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Plan Your Packaging Strategy

At this point in the design process, differentiate which features are available at a user-level vs. org-level. For user-level features, your customers purchase and assign a license for each user who needs access. For org-level features, your customers can purchase one product and all of their users, or their whole Salesforce org, have access to that feature.

For example, say you're developing a product called Travel Navigation, which includes:

- A basic set of functionality, such as being able to view, create, and edit visual maps with custom labels.
- An add-on for enhanced functionality, such as being able to color code labels and add symbols.
- An add-on for functionality around territories, which allows creating visual territories and subterritories.

You plan to sell each of these three sets of functionalities separately on a per-user basis.

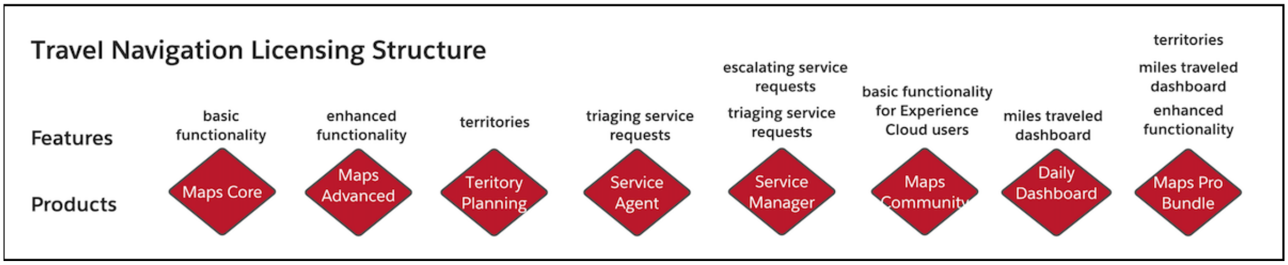
Your product also has a miles traveled dashboard functionality, which shows your customer how many miles in a given day that a sales or service person travels. You plan to sell this dashboard at the org-level.

Identify Products

Next, map the features to the products that you want to sell and that entitle the customer to those features. For each of the distinct features that you define in your pricing and feature strategy, you can have one or more products that contain a feature. You can also have more than one feature in one product.

For our Travel Navigation example, you create this matrix to outline your product mapping.

Feature	Relationship	Product
basic functionality	contained in	Maps Core
enhanced functionality	contained in	Maps Advanced
territories	contained in	Territory Planning
triaging service requests	contained in	Service Agent
triaging service requests	contained in	Service Manager
escalating service requests	contained in	Service Manager
basic functionality for Experience Cloud users	contained in	Maps Community
miles traveled dashboard	contained in	Daily Dashboard
territories	contained in	Maps Pro Bundle
miles traveled dashboard	contained in	
enhanced functionality	contained in	



Note that the triaging service requests feature is contained in both Service Agent and Service Manager products and that Service Manager contains two features.

We recommend using a visual diagramming tool to build your licensing design.

SEE ALSO:

Licensing Components (Developer Preview)

Identify Licenses (Developer Preview)

After you develop your feature pricing strategy, identify the licenses that each product contains. Your licenses contain the underlying features that your products entitle.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Consider what custom permission set licenses you must create. Custom permission set licenses are assigned to users and give the entitlements contained within the custom permission set license to the users.

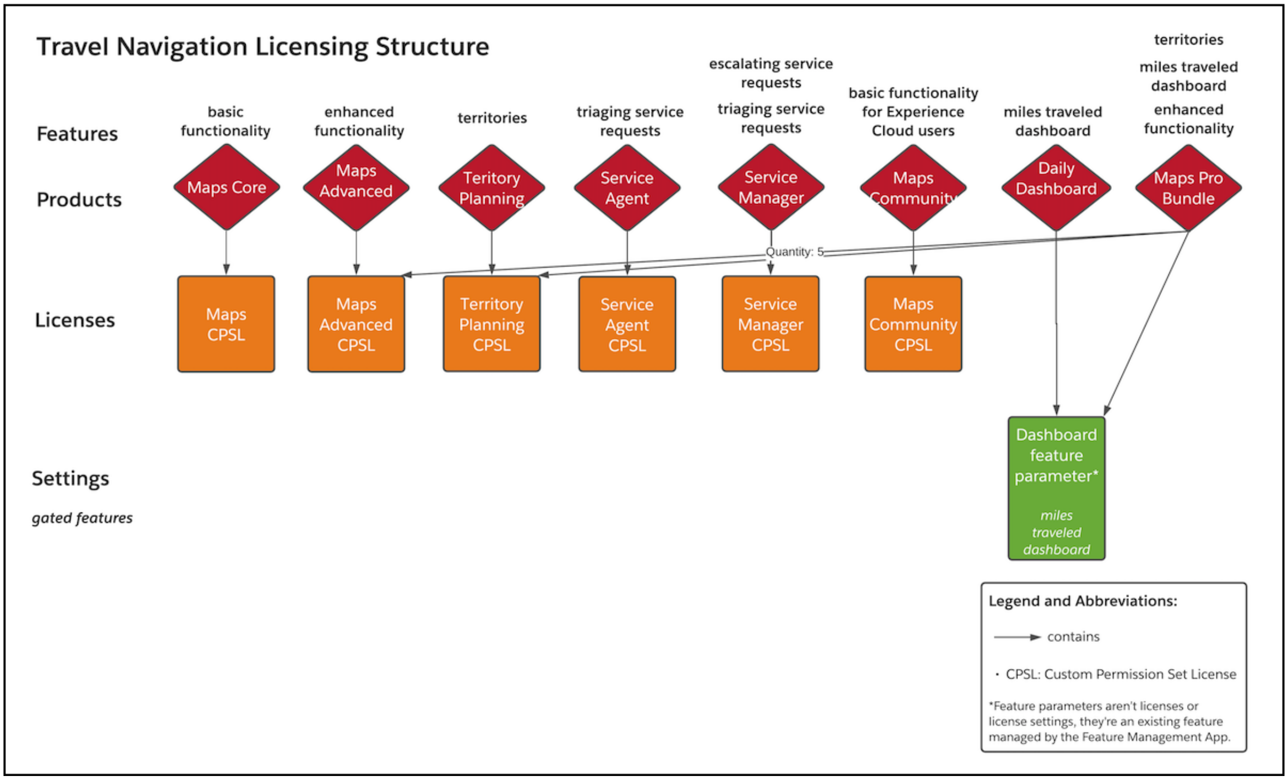
Next, consider how to include your set of licenses into your set of products. When a product is purchased by your customers, you'll provision the underlying licenses to their org. The customer then assigns these licenses and permission sets to users to access the functionality entitled by the license.

For Travel Navigation, you create this matrix to outline the relationship between your products and custom permission set licenses (CPSLs).

Product	Relationship	License
Maps Core	contains 1	Maps CPSL
Maps Advanced	contains 1	Maps Advanced CPSL
Territory Planning	contains 1	Territory Planning CPSL
Service Agent	contains 1	Service Agent CPSL
Service Manager	contains 1	Service Manager CPSL
Maps Community	contains 5	Maps Community CPSL
Daily Dashboard	contains 1	Dashboard feature parameter*
Maps Pro Bundle	contains 1	Dashboard feature parameter*

Product	Relationship	License
	contains 1	Territory Planning CPSL
	contains 5	Maps Advanced CPSL

*Feature parameters aren't licenses. They are an existing feature and are managed by the Feature Management App.



Note that the Maps Pro Bundle product contains multiple licenses, and some individual licenses are contained within multiple products. This many-to-many mapping of licenses and products is common.

SEE ALSO:

- Manage Features
- Licensing Components (Developer Preview)

Identify License Settings Use Cases (Developer Preview)

Identify the correct settings to include in your licenses that gate specific functionality so that users have the access to the intended features.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

In this table, we summarize the licensing mechanism for each type of access for your product.

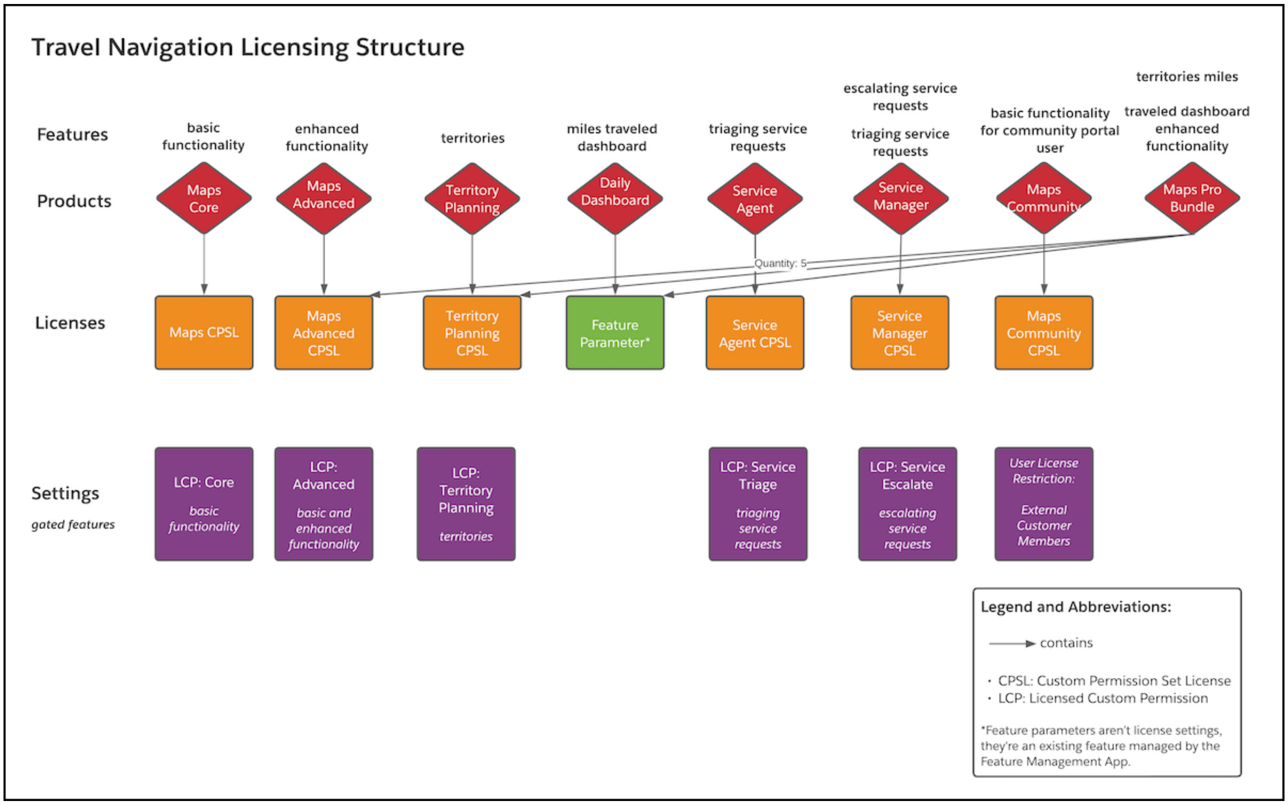
	Feature Access	Object Access	Package Access	Usage Entitlement
User-Level	Licensed Custom Permission	*In Consideration*	Managed Package License or Custom Foundation Permission Set License	Not Applicable
Org-Level	LMO-to-Subscriber Boolean Feature Parameter*	Custom Object	Site-wide license setting in LMA	Subscriber-to-LMO Integer Feature Parameter (usage metric) and LMO-to-Subscriber Integer Feature Parameter (usage limit) pairing*

*Feature parameters aren't license settings. They are an existing feature and are managed by the Feature Management App.

Continuing our Travel Navigation example, we create this matrix to capture the exact settings and licensed custom permissions (LCPs), and their relation to your identified features.

Setting	Relationship	Feature
Custom Foundation Permission Set License	gates	overall package namespace access
LCP: Core	gates	basic functionality
LCP: Advanced	gates	enhanced functionality
LCP: Territory Planning	gates	territories
LCP: Service Triage	gates	triaging service requests
LCP: Service Escalate	gates	escalating service requests
LCP: Core and User License Restrictions: External Customer Members	gates	basic functionality for Experience Cloud users
Dashboard feature parameter*	gates	miles traveled dashboard

*Feature parameters aren't license settings. They are an existing feature and are managed by the Feature Management App.


Travel Navigation Licensing Structure
Features
Products
Licenses
Settings
gated features
basic functionality
enhanced functionality
territories
miles traveled dashboard
triaging service requests
escalating service requests
triaging service requests
basic functionality for community portal user
territories miles traveled dashboard enhanced functionality
Maps Core
Maps Advanced
Territory Planning
Daily Dashboard
Service Agent
Service Manager
Maps Community
Maps Pro Bundle
Quantity: 5
Maps CPSL
Maps Advanced CPSL
Territory Planning CPSL
Feature Parameter*
Service Agent CPSL
Service Manager CPSL
Maps Community CPSL
LCP: Core
basic functionality
LCP: Advanced
basic and enhanced functionality
LCP: Territory Planning
territories
LCP: Service Triage
triaging service requests
LCP: Service Escalate
escalating service requests
User License Restriction:
External Customer Members
Legend and Abbreviations:
contains
CPSL: Custom Permission Set License
LCP: Licensed Custom Permission
*Feature parameters aren't license settings, they're an existing feature managed by the Feature Management App.

For examples of gating access in other pricing scenarios, refer to these topics.

User-Level Feature Gating (Developer Preview)
When you're ready to design your user-level license settings, review these common scenarios and suggested solutions.

Org-Level Feature Gating (Developer Preview)
When you're ready to design your org-level licensing strategy, review these common scenarios and suggested solutions.

Advanced Use Cases and Compound Gating (Developer Preview)
Review advanced pricing scenarios that require compound solutions or specific configurations.

SEE ALSO:

Licensing Components (Developer Preview)

Drive App Behavior with LMO-to-Subscriber Feature Parameters

Track Preferences and Activation Metrics with Subscriber-to-LMO Feature Parameters

User License Restriction Categories (Developer Preview)

User-Level Feature Gating (Developer Preview)

When you're ready to design your user-level license settings, review these common scenarios and suggested solutions.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Requirements	Solution
You want to sell a feature via a license that must be bought and assigned individually for each user.	Licensed custom permissions. These permissions are checked in your Apex code during runtime. The user is given access to the gated feature if the user has a custom permission set license containing the licensed custom permission and a permission set assigned that grants them the licensed custom permission.
You want to sell access to your whole package via a license that must be bought and assigned individually for each user. You're migrating away from managed package licenses, which perform a similar function.	A custom foundation permission set license. This license gives the user the ability to enter the package namespace and the entitlement to access all foundation features, similar to managed package licenses.
You want to sell a feature that your subscriber can purchase separately for their standard internal Salesforce users and external Experience Cloud site users. For example, you want to sell a product and license for internal users for $20 per user per month and a similar but separate product and license for external users for $2 per user per month. You want to ensure that the subscriber doesn't use the cheaper license for a more expensive type of user.	User license restrictions specified in two custom permission set licenses for each user category.

SEE ALSO:

- Identify License Settings Use Cases (Developer Preview)
- Licensing Components (Developer Preview)
- Org-Level Feature Gating (Developer Preview)
- Advanced Use Cases and Compound Gating (Developer Preview)

Org-Level Feature Gating (Developer Preview)

When you're ready to design your org-level licensing strategy, review these common scenarios and suggested solutions.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Requirements	Solution
You want to sell a feature via a product that must be bought only one time to give all users in the Salesforce org access to the feature.	Feature parameter of type boolean. This is access checked in your Apex code during runtime and gives any user in the org access to the gated feature. No permission set or custom permission set license assignment is required for a user to access the feature. Note: Feature parameters are managed by the Feature Management App. The Feature Management App is an

Requirements	Solution
	existing feature that is separate from the Partner Licensing Platform.
You want to sell access to your whole package via a product that must be bought only one time to give all users in the Salesforce org access to your whole package.	Use the License Management App to set licensing for a particular subscriber to be site-wide. Note: The License Management App is an existing feature that is separate from the Partner Licensing Platform. Note that permission set assignments for various permissions, such as custom object permissions, are still required for users in the org to access those features.
You want to sell a consumption-based entitlement, such as a number of API calls or call center minutes, in a specific quantity for the entire Salesforce org. This quantity is consumed by any user accessing this entitlement in the org.	At a high level, this requirement can be achieved via a LMO-to-Subscriber feature parameter to track the total sold quantity and a Subscriber-to-LMO feature parameter for tracking the used quantity from your package code. There may be other considerations depending on your specific needs, such as billing, usage periods, overages, and restricting or allowing access when the total sold quantity is consumed. Note: Feature parameters are managed by the Feature Management App. The Feature Management App is an existing feature that is separate from the Partner Licensing Platform.

SEE ALSO:

Identify License Settings Use Cases (Developer Preview)

Licensing Components (Developer Preview)

User-Level Feature Gating (Developer Preview)

Advanced Use Cases and Compound Gating (Developer Preview)

Advanced Use Cases and Compound Gating (Developer Preview)

Review advanced pricing scenarios that require compound solutions or specific configurations.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Example: Org-Level Feature with User-Level Licenses

Requirement: You want to sell access to a feature via a product that must be bought one time for the org and a license that must be bought and assigned individually for each user. The Salesforce org must have both the org-level product and the user-level license assigned for a user to be able to access the feature.

For example, an intelligence dashboard, which you want to sell for $200/month for the org to have access as the standard price, and then an incremental $25/user/month for each user to access the dashboard.

Solution: A combination of a boolean feature parameter and a licensed custom permission. Both the feature parameter and licensed custom permission are access checked in Apex code during runtime. If the user has a custom permission set license containing the licensed custom permission and the org has the feature parameter turned on, the user gets access to the feature. Remember that the user must also have a permission set assigned that grants them access to the licensed custom permission.

Note: Feature parameters are managed by the Feature Management App. The Feature Management App is an existing feature that is separate from the Partner Licensing Platform.

Example: Org-Level Feature with Admin-Controlled Access

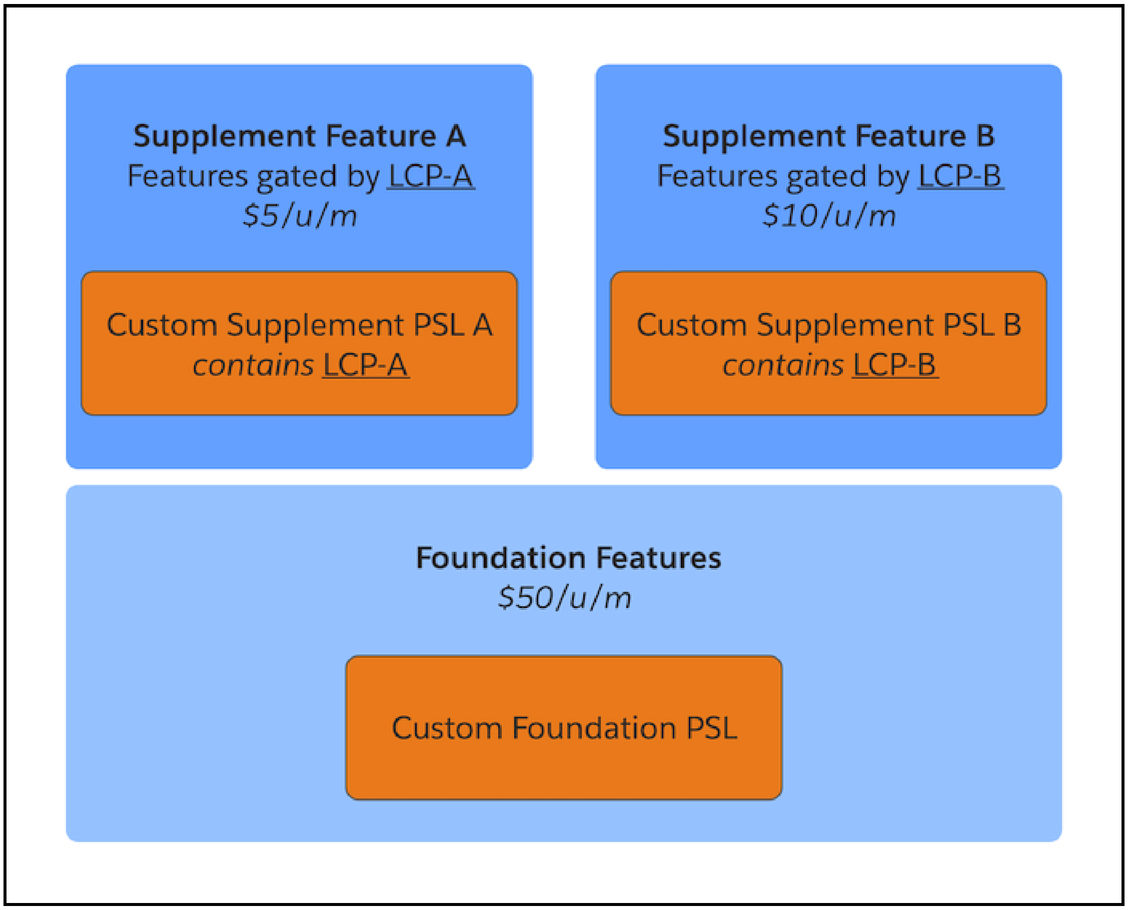
Requirement: You want to sell access to a product that must be bought one time for the Salesforce org and no further licenses are needed for each user. However, you want to ensure that the org admin decides which users get access to this feature.

Solution: Use a boolean feature parameter to gate access to the feature for an org, and then a regular custom permission to gate access for the user. Both the feature parameter and custom permission are access checked in Apex code during runtime to give access to the feature. Note that the user doesn't require a license for the feature and the org admin can assign the user a permission set that grants them the custom permission.

Example: Foundation and Supplement Features

Requirement: You want to sell access to foundation features in one license, and then sell supplement features in supplement licenses. You want to ensure that the supplement licenses don't give access to the foundation features and that a user must first be assigned the foundation license to receive the supplement licenses.

For example, you sell access to foundation features and your package namespace for $50 per user per month. Then, you sell supplement feature A for $5 per user per month, and supplement feature B for $10 per user per month. You want to ensure that the customer doesn't get access to the foundation features and your overall package only by assigning the cheaper supplement license.



Solution: Create a custom foundation permission set license for your foundation features. This license entitles access to all your foundation features, including your package namespace, and doesn't entitle access to the supplement features A or B.

Then, create custom supplement permission set license A containing licensed custom permission A, and custom supplement permission set license B containing licensed custom permission B. Your feature A and feature B are gated by licensed custom permissions A and B in Apex code, respectively. Since these two licenses are supplement licenses, they don't give access to your foundation features, including your package namespace.

Since your custom supplement permission licenses don't give access to your foundation features, a user must first have your custom foundation permission set license assigned in order to receive either or both of the custom supplement permission set licenses. This setup ensures that the customer can't assign only a cheaper supplement license and get access to all the foundation features, including your package namespace.

Additionally, users sometimes only require access to the foundation features and not the additional supplement features. These users can be assigned only the custom foundation permission set license and not the supplement licenses.

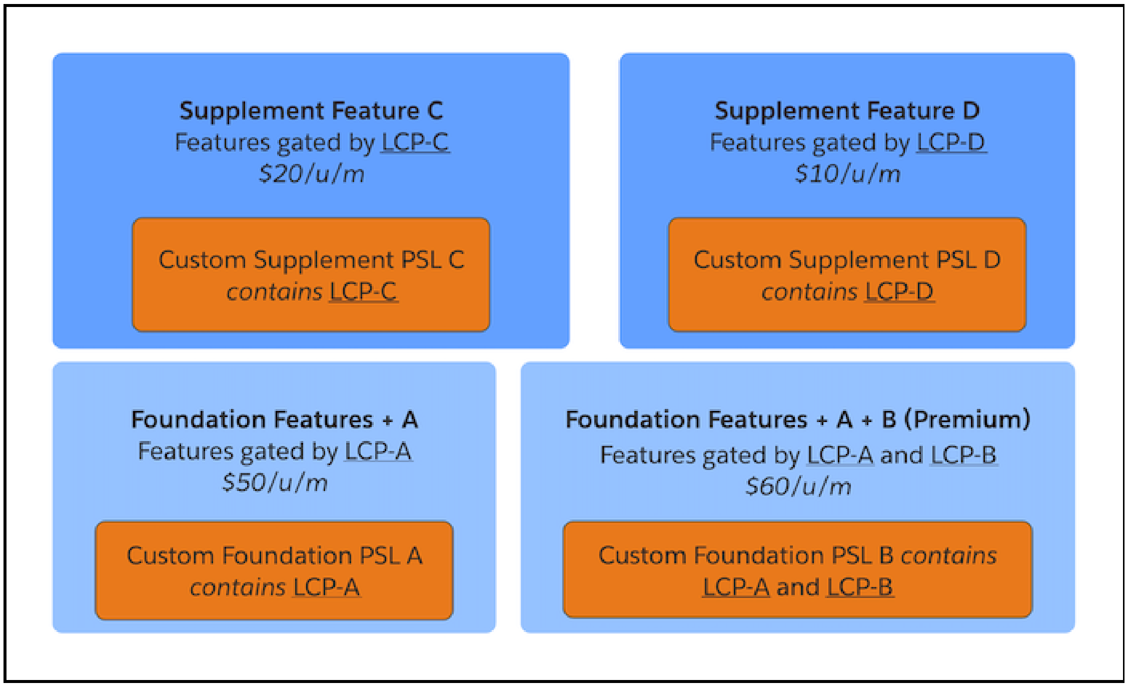
Example: Multiple Foundation and Supplement Features

Requirement: You want to sell two versions of foundation licenses, where every user of your package must have at least one of the two foundation licenses assigned. One of the foundation licenses contains an additional feature A and the other contains additional features A and B.

Then, you also want to sell supplement features in supplement licenses. You want to ensure that the supplement licenses don't give access to the foundation features and that a user must first be assigned a foundation license to receive the supplement licenses.

For example, you sell access to the foundation features, your package namespace, and feature A for $50 per user per month. You also sell access to the foundation features, your package namespace, and features A and B for $60 per user per month. These two licenses are only differentiated by which additional features they entitle, and are considered as two "versions" of your product, one being more premium with an additional feature.

Then, you sell supplement feature C for $20 per user per month, and supplement feature D for $10 per user per month. You want to ensure that the customer doesn't get access to the foundation features and your overall package only by assigning the cheaper supplement license.



Solution: Create a custom foundation permission set license A containing licensed custom permission A. Create a second custom foundation permission set license B containing licensed custom permissions A and B. Your feature A and feature B are gated by licensed custom permissions A and B in Apex code, respectively. These licenses entitle access to all your foundation features, including your package namespace, and either feature A or features A and B. They don't entitle access to the supplement features C or D.

Then, create custom supplement permission set license C containing licensed custom permission C, and custom supplement permission set license D containing licensed custom permission D. Your feature C and feature D are gated by licensed custom permissions C and D in Apex code, respectively. Since these two licenses are supplement licenses, they don't give access to your foundation features, including your package namespace.

Since your supplement licenses don't give access to your foundation features, a user must first have one of your custom foundation permission set licenses assigned in order to receive either or both of the custom supplement permission set licenses. This setup ensures that the customer can't assign only a cheaper supplement license and get access to all the foundation features, including your package namespace.

Additionally, users sometimes only require access to the foundation features and not the supplement features. These users can be assigned only one of the custom foundation permission set licenses and not the supplement licenses.

An alternative approach is that you could consider having only the licensed custom permission B in your premium custom foundation permission set license. In this case, you would gate access to feature A with either licensed custom permission A or B. This way, the user with the premium custom foundation permission set license would get access to both features A and B with only one "premium" licensed custom permission B. This may be desired if you'd like to present licensed custom permission B as giving access to both features or make it simpler for your customer admins to authorize both features with one permission. This example illustrates the flexibility of licensing and access check capabilities to meet your desired feature pricing and access strategy.

Important: Be deliberate when creating foundation vs. supplement licenses for your package features. Most often, you'll create foundation licenses, as these licenses allow the user to enter the package namespace and are a replacement for the original managed package licenses. Supplement licenses are usually used for additional functionality that you want to sell on top of your foundation features, as these licenses require the user to have at least one foundation license assigned first.

SEE ALSO:

- Identify License Settings Use Cases (Developer Preview)
- Licensing Components (Developer Preview)
- User-Level Feature Gating (Developer Preview)
- Org-Level Feature Gating (Developer Preview)

Map Settings to Licenses (Developer Preview)

After you confirm a list of license settings for your features, decide which settings to include in which custom permission set licenses.

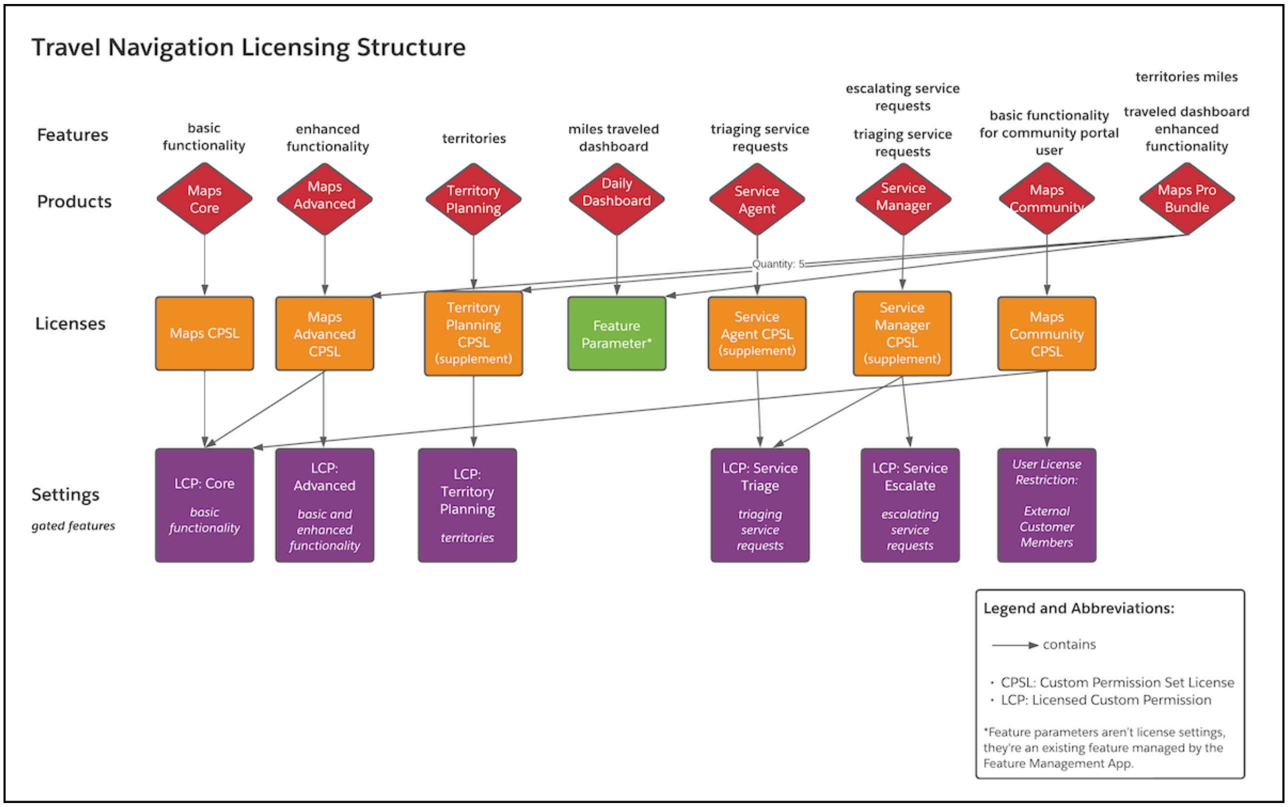
Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn’t generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Licenses can contain multiple settings, and one setting can be contained in multiple licenses.

For the Travel Navigation product, you create this matrix to match up your settings, or licensed custom permissions (LCPs), to your custom permission set licenses. You also specify which licenses are foundation licenses, which provide access to the package namespace and foundation features, and which are supplement licenses, which only entitle users to additional features, not the package overall.

License	Relationship	Setting
Maps (foundation license)	contains	LCP: Core
Maps Advanced (foundation license)	contains	LCP: Core
		LCP: Advanced
Territory Planning (supplement license)	contains	LCP: Territory Planning
Service Agent (supplement license)	contains	LCP: Service Triage
Service Manager (supplement license)	contains	LCP: Service Triage
		LCP: Service Escalate
Maps Community (foundation license)	contains	LCP: Core and User License Restriction: External Customer Members
Daily Dashboard (product, not license)	contains	Dashboard feature parameter*

*Feature parameters aren’t license settings. They are an existing feature and are managed by the Feature Management App.


Travel Navigation Licensing Structure
Features
Products
Licenses
Settings
gated features
basic functionality
enhanced functionality
territories
miles traveled dashboard
triaging service requests
escalating service requests
triaging service requests
basic functionality for community portal user
territories miles traveled dashboard enhanced functionality
Maps Core
Maps Advanced
Territory Planning
Daily Dashboard
Service Agent
Service Manager
Maps Community
Maps Pro Bundle
Quantity: 5
Maps CPSL
Maps Advanced CPSL
Territory Planning CPSL (supplement)
Feature Parameter*
Service Agent CPSL (supplement)
Service Manager CPSL (supplement)
Maps Community CPSL
LCP: Core
basic functionality
LCP: Advanced
basic and enhanced functionality
LCP: Territory Planning
territories
LCP: Service Triage
triaging service requests
LCP: Service Escalate
escalating service requests
User License Restriction:
External Customer Members
Legend and Abbreviations:
contains
CPSL: Custom Permission Set License
LCP: Licensed Custom Permission
*Feature parameters aren't license settings, they're an existing feature managed by the Feature Management App.

Note: The Territory Planning, Service Agent, and Service Manager licenses are custom supplement permission set licenses that must be assigned to a user in conjunction with a custom foundation permission set license to be functional.

SEE ALSO:

Licensing Components (Developer Preview)

User License Restriction Categories (Developer Preview)

Design Permission Sets (Developer Preview)

The design of permission sets is based on your intended end-user personas and the appropriate access they require to perform their tasks. The goal is to enable your customers to have a convenient administration experience.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

So far, you've completed the bulk of your licensing design. You have a set of products, licenses, and license settings that will **entitle** your package features according to your feature pricing and licensing strategy. Now, it's time to design your permission sets to enable your customers to easily **grant access** to your package features out of the box.

Your customers often have multiple other products with their own licenses and permission sets. Giving your customers easy-to-understand and out-of-box personas packaged as custom permission sets enables them to administer and onboard onto your product faster.

Permission Sets

First, review your customer personas. Think through what types of user personas will access your product functionality, what features of your package they'll need access to, and in what capacity.

Next, design the permission sets for these personas. Think of each persona as having one permission set that gives them the required access. In many cases, you can define one permission set that contains all the permissions for the features you intend for a persona.

Keep in mind that your customer admin can always create custom permission sets in their org and assign any combination of permissions from your licenses as well as Salesforce licenses. However, such a permission set can only be assigned to a user who has all the necessary underlying licenses assigned to them.

As a best practice, make sure that no permission set contains more permissions than what its intended, related custom permission set license contains. If one permission set contains permissions from multiple custom permission set licenses, break up that permission set into multiple permission sets. Then use a permission set group to group those permission sets together. This practice ensures that each permission set can be associated with a single custom permission set license, such that it grants some or all of the entitlements from that license, and not more. Also note there can be multiple permission sets that contain license settings from one custom permission set license.

For our Travel Navigation example, you've identified these personas:

Persona	Relationship	Feature
Org Manager	can access	enhanced functionality
		territories
		triaging service requests
		escalating service requests
Service Agent	can access	triaging service requests
Service Manager	can access	triaging service requests
		escalating service requests
Service Support	can access	escalating service requests
Territory Planner	can access	territory planning

Here's one process by which you can design the permission sets for these personas.

1. Create one permission set, which contains one or more licensed custom permissions (LCPs), for each custom permission set license. Then, map them to each other.

Settings/Permission	Relationship	Permission Set	Relationship	Custom Permission Set License
LCP: Core	is contained in	Maps Core	maps to	Maps
LCP: Advanced	is contained in	Maps Advanced	maps to	Maps Advanced
LCP: Territory Planning	is contained in	Territory Planning	maps to	Territory Planning
LCP: Service Triage	is contained in	Service Agent	maps to	Service Agent
LCP: Service Triage	is contained in	Service Manager	maps to	Service Manager

Settings/Permission	Relationship	Permission Set	Relationship	Custom Permission Set License
LCP: Service Escalate				

2. Consider if a particular persona needs only a subset of the access from a custom permission set license. Let's say that you have a custom permission set license that maps to four permissions, but the persona only needs two of them. Create another permission set with the limited access.

   In this example, the Service Support persona doesn't need all the permissions licensed by the Service Manager custom permission set license, and similarly needs less access than the Service Manager persona. So we add the following permission set:

Settings/Permission	Relationship	Permission Set	Relationship	Custom Permission Set License
LCP: Service Escalate	is contained in	Service Support	maps to	Service Manager

Permission Set Groups

A permission set group streamlines permissions assignment and management by grouping together multiple permission sets based on user job functions. Users assigned the permission set group receive the combined permissions of all the permission sets in the group.

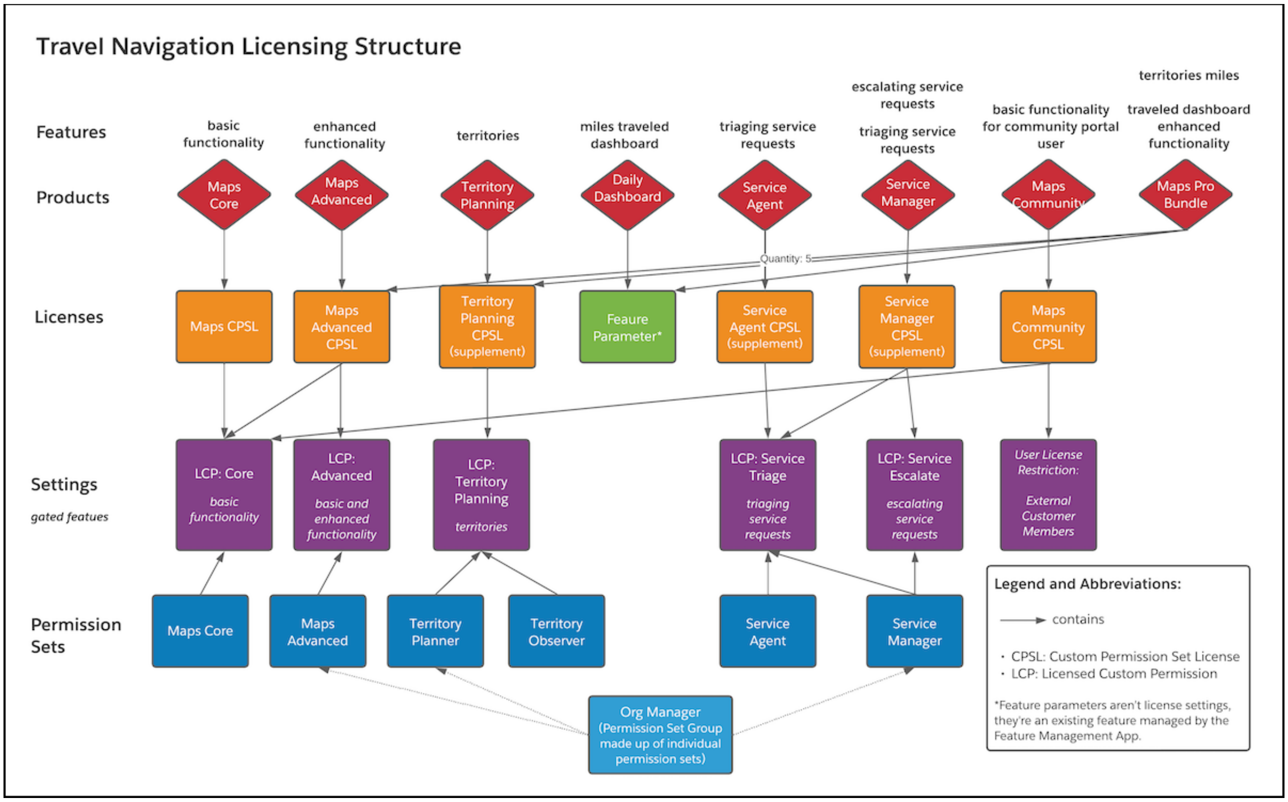
In our example, we check if a particular persona needs a superset of access from more than one permission set, and group those permission sets into a permission set group. The Org Manager overseeing both the maps and service teams must access the advanced maps functionality, territories functionality, and all service agent functionality. Note that the feature access of this persona crosses over multiple custom permission set licenses.

We create the following permission set group:

Permission Set Group	Relationship	Permission Set	Relationship	Settings/Permission
Org Manager	contains	Maps Advanced	contains	LCP: Advanced
		Territory Planner	contains	LCP: Territory Planning
		Service Manager	contains	LCP: Service Triage
				LCP: Service Escalate

Remember that assigning the Org Manager permission set group to a user grants the user all permissions from the underlying permission sets. The permission sets themselves aren't directly assigned on their own. If subscriber admins don't want to grant every permission from the permission set group, they can mute a specific permission in the group with a muting permission set.

Finally, assess whether any permission set is unnecessary, meaning the personas you designed are getting the access they need through another permission set or permission set group.


Travel Navigation Licensing Structure
Features
Products
Licenses
Settings
gated featues
Permission Sets
basic functionality
enhanced functionality
territories
miles traveled dashboard
triaging service requests
escalating service requests
triaging service requests
basic functionality for community portal user
territories miles traveled dashboard enhanced functionality
Maps Core
Maps Advanced
Territory Planning
Daily Dashboard
Service Agent
Service Manager
Maps Community
Maps Pro Bundle
Quantity: 5
Maps CPSL
Maps Advanced CPSL
Territory Planning CPSL (supplement)
Feaure Parameter*
Service Agent CPSL (supplement)
Service Manager CPSL (supplement)
Maps Community CPSL
LCP: Core
basic functionality
LCP: Advanced
basic and enhanced functionality
LCP: Territory Planning
territories
LCP: Service Triage
triaging service requests
LCP: Service Escalate
escalating service requests
User License Restriction:
External Customer Members
Maps Core
Maps Advanced
Territory Planner
Territory Observer
Service Agent
Service Manager
Org Manager (Permission Set Group made up of individual permission sets)
Legend and Abbreviations:
contains
• CPSL: Custom Permission Set License
• LCP: Licensed Custom Permission
*Feature parameters aren't license settings, they're an existing feature managed by the Feature Management App.

Design Notes

- The Maps Advanced custom permission set license includes both Core and Advanced licensed custom permissions, but the Maps Advanced permission set only includes the Advanced permission. This structure implies that both the basic Core functionality and enhanced Advanced functionality can be accessed by the Advanced licensed custom permission.

  The Apex code gating those two functionalities accounts for this design choice. It access checks for either the Core or Advanced licensed custom permission when gating basic functionality, and access checks only the Advanced permission when gating enhanced functionality.

  Alternatively, you could include both Core and Advanced licensed custom permissions in the Maps Advanced permission set, and access check basic and enhanced functionality separately with Core and Advanced licensed custom permissions, respectively.

- A user needing only basic functionality can still be assigned the Maps Advanced permission set license. Then, the user is assigned a permission set with the Core licensed custom permission only so that they're granted access only to basic functionality while having the Maps Advanced permission set license.
- In this example, an Experience Cloud functionality permission set isn't designed. Instead, we're using the External Customer Member user license restriction category to control who can be assigned the permission set license.

Note: There are many ways by which you can design your permission sets. This topic is a guideline to help you think through your personas, but you can design your permission sets in a different way. Ultimately, ease of administration and out-of-box configuration for your customers is the goal.

SEE ALSO:

- Licensing Components (Developer Preview)
- Entitlements and Grants (Developer Preview)
- *Salesforce Help:* Permission Sets
- *Salesforce Help:* Permission Set Groups

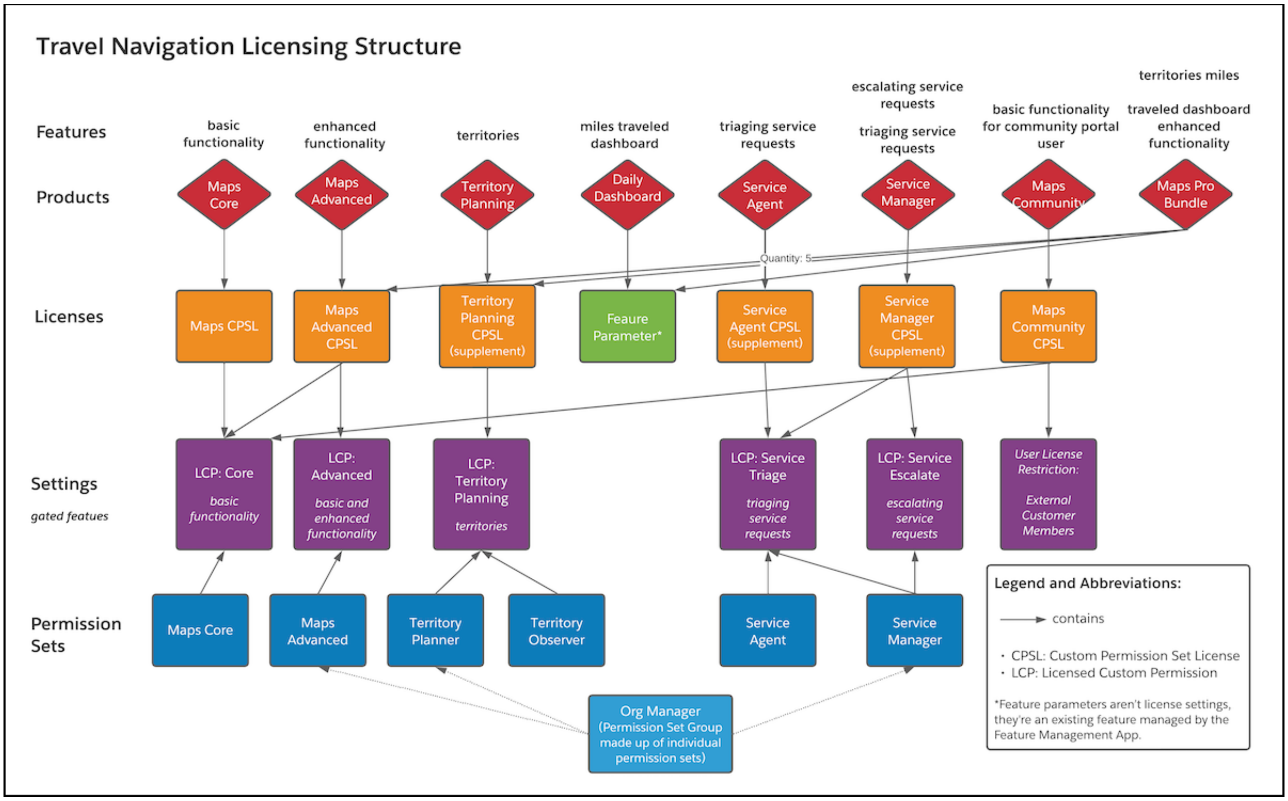
Review Your Structure (Developer Preview)

Review your design to ensure that your licensing and permissions are structured as you need for your feature pricing and licensing strategy.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

What a journey! By this point, you've designed your whole feature pricing and licensing structure. You first designed products, licenses, and licensed settings. Next, you mapped settings to licenses, and finally designed the permission sets for your personas.

Here's the final result of this exercise for the Travel Navigation example again.


Travel Navigation Licensing Structure
Features
Products
Licenses
Settings
gated featues
Permission Sets
basic functionality
enhanced functionality
territories
miles traveled dashboard
triaging service requests
escalating service requests
triaging service requests
basic functionality for community portal user
territories miles
traveled dashboard enhanced functionality
Maps Core
Maps Advanced
Territory Planning
Daily Dashboard
Service Agent
Service Manager
Maps Community
Maps Pro Bundle
Quantity: 5
Maps CPSL
Maps Advanced CPSL
Territory Planning CPSL (supplement)
Feaure Parameter*
Service Agent CPSL (supplement)
Service Manager CPSL (supplement)
Maps Community CPSL
LCP: Core
basic functionality
LCP: Advanced
basic and enhanced functionality
LCP: Territory Planning
territories
LCP: Service Triage
triaging service requests
LCP: Service Escalate
escalating service requests
User License Restriction:
External Customer Members
Maps Core
Maps Advanced
Territory Planner
Territory Observer
Service Agent
Service Manager
Org Manager (Permission Set Group made up of individual permission sets)
Legend and Abbreviations:
contains
• CPSL: Custom Permission Set License
• LCP: Licensed Custom Permission
*Feature parameters aren't license settings, they're an existing feature managed by the Feature Management App.

Before you implement your design, we encourage you to review it thoroughly. While feedback from the implementation and testing phases can inform your design via iteration, it's harder to restructure your design later on.

During the design process, it's common to think of new scenarios and change your design. Make sure to think through various use cases for your customers. Also consider potentially faulty licensing structures, such as over-permissioning or under-permissioning, which can allow your subscriber to get more access without paying or receive less access than they expect. After your implementation phase, you'll want to establish a thorough test plan to ensure that everything works as expected.

In this design guide, we walked you through a "top-down" approach, starting from main features and overall pricing strategy, and working through to the individual components. This exercise can also be performed the other way around, or "bottom-up": starting with the individual license settings and mapping them through permission sets, licenses, products, and then your overall pricing strategy. It may be helpful to think through your design this way to surface additional considerations.

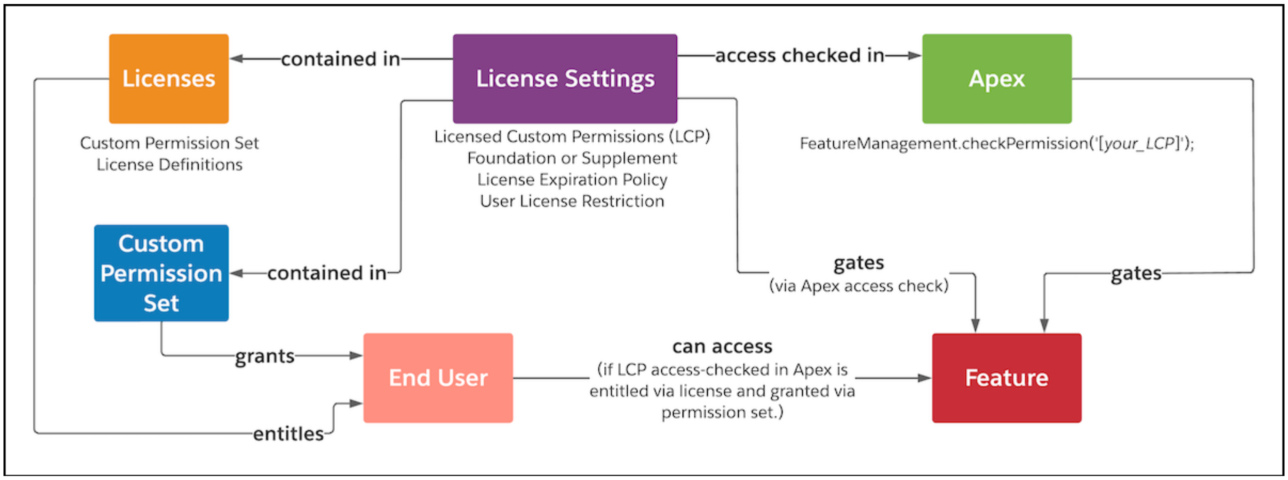
Implement Your Package Licensing (Developer Preview)

After completing your licensing and permissioning design, you're ready to create your licensing components using point-and-click tools and Apex.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

In addition to this guidance, we prepared a demo application that you can reference as you implement your package licensing, or simply use to deploy and test the new functionality.

Create your licensing components in a development scratch org created from your new Dev Hub that is enabled with the Partner Licensing Platform. To enable the Partner Licensing Platform in this org, make sure that you have the `PartnerLicensingPlatform` feature in your scratch org definition file.


Licenses
contained in
License Settings
access checked in
Apex
Custom Permission Set License Definitions
Licensed Custom Permissions (LCP)
Foundation or Supplement
License Expiration Policy
User License Restriction
FeatureManagement.checkPermission('[your_LCP]');
Custom Permission Set
contained in
gates
(via Apex access check)
gates
grants
End User
can access
(if LCP access-checked in Apex is entitled via license and granted via permission set.)
Feature
entitles

You first create licensed custom permissions in your development scratch org, then put these custom permissions into custom permission set license definitions, which entitle the custom permissions. Next, you put the custom permissions into custom permission sets, which grant the custom permissions. Finally, you put access checks for the licensed custom permissions in your Apex code for specific features. The access checks gate access to the features. From there, you're ready to deploy and create your new package.

Create Licensed Custom Permissions (Developer Preview)

Licensed custom permissions gate access to your package features.

Define Custom Permission Set Licenses (Developer Preview)

The custom permission set license contains your package's licensed custom permissions and provides the entitlement for the features gated by those settings.

Create Custom Permission Sets (Developer Preview)

The permission set provides the grant to access the package features already entitled by a license.

Create Custom Permission Set Groups (Developer Preview)

Permission set groups bundle permission sets together based on user job functions. Create permission set groups if your licensing structure requires a persona to have permissions from multiple custom permission set licenses.

Create Access Checks in Apex (Developer Preview)

Create access checks for your licensed custom permissions in your Apex code to gate access to your features.

Deploy Your Licensing Structure and Create Your Package (Developer Preview)

After you implement your new licenses and settings, you're ready to create a new package with this metadata.

Create Licensed Custom Permissions (Developer Preview)

Licensed custom permissions gate access to your package features.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Create your licensed custom permissions in a scratch org created from your new Dev Hub.

1. From Setup, in the Quick Find Box, enter `Custom Permissions`.
2. Select **Custom Permissions**.
3. Click **New**.
4. Enter a label. The Name field is generated automatically.
5. Enter a description.
6. Select **License Required**. This setting indicates that this permission can be granted to a user only if that user also has a custom permission set license containing this licensed custom permission.

   Note: After a custom permission is packaged and released, you can't edit this setting.

7. Click **Save**.

SEE ALSO:

Salesforce Help: Custom Permissions

Partner Licensing Platform Components and Concepts (Developer Preview)

Identify Licenses (Developer Preview)

Define Custom Permission Set Licenses (Developer Preview)

The custom permission set license contains your package's licensed custom permissions and provides the entitlement for the features gated by those settings.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Create your custom permission set license definitions in a scratch org created from your new Dev Hub. Make sure to review the Licensing Components topic before creating your license definitions.

Note: Custom permission set licenses are provisioned and assignable license seats on the subscriber side. On the developer side, they're called permission set license **definitions**, as they're the package metadata, not the actual seats of the license.

1. In your scratch org, from Setup, in the Quick Find Box, enter `Custom Permission Set License Definitions`.
2. Select **Custom Permission Set Licenses Definitions**.
3. Click **New**.
4. Enter a label. The Name field is generated automatically.
5. Enter a description.
6. For License Expiration Policy, indicate whether access to the package is blocked for assigned users when all custom permission set licenses expire.
7. If this license is a supplement license, select **Supplement license**.

   Note: This setting can't be edited after the license is created. Supplement licenses don't give access to the package's foundation features, including the package namespace.

8. Select which users can be assigned this permission set license using the user license restriction categories. By default, when no categories are selected, all user licenses can be assigned the permission set license. After you select one user license category, only users with licenses contained in the selected categories can be assigned the permission set license.

   Note: For which user licenses are included in each category, see User License Restriction Categories (Developer Preview).

9. Click **Save.**
10. Under Licensed Custom Permissions, click **Add Licensed Custom Permission**.
11. Select the permission that you created previously and click **Save**.

    Note: You can edit the permissions that you include in your license and upload a new version. However, if you remove permissions, this change only affects new customers. Existing customers retain the same entitlements as before.

User License Restriction Categories (Developer Preview)
Review which user licenses are included in each user license restriction category. These categories are used to specify which users can be assigned your custom permission set licenses.

SEE ALSO:

Partner Licensing Platform Components and Concepts (Developer Preview)

Identify Licenses (Developer Preview)

Entitlements and Grants (Developer Preview)

User License Restriction Categories (Developer Preview)

Review which user licenses are included in each user license restriction category. These categories are used to specify which users can be assigned your custom permission set licenses.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Note: Some of these licenses are supported but are no longer available for purchase.

Category	Category API Value	User Licenses
All Internal Users	`${internal}`	• Analytics Cloud Integration User • B2BMA Integration User • Chatter External • Chatter Free • Chatter Only • Company Communities • Content Only • CPQ Integration User • Cross-Org Usages • Cross Org Data Proxy • Database.com Admin • Database.com Light • Database.com User • Developer • Force.com - App Subscription • Force.com - Free • Force.com - One App • Ideas Only • Identity • Knowledge Only • Licensing API

Category	Category API Value	User Licenses
		- Partner App Subscription - Partner Network - Premier Support - Sales Insights Integration User - Salesforce - Salesforce Limited Access - Free - Salesforce Platform - Salesforce Platform Light - Salesforce Platform One - SalesforceIQ Integration User - Service Cloud - Service User CMT Query Metrics - Service User Deep Sea Token Access - Service User DI Cross Org Access - Service User EDW Bulk Query - Service User GPU CMF Integration - Siteforce Only - Work.Com Only
Internal Platform Users	`${platform}`	- Chatter Only - Company Communities - Force.com - App Subscription - Force.com - Free - Force.com - One App - Identity - Partner App Subscription - Salesforce Platform - Salesforce Platform Light - Salesforce Platform One - Siteforce Only
All External Customer and Partner Members	`${communities}`	- Authenticated Website - Bronze Partner - Channel Account - Chatter External - Company Communities - Customer Community - Customer Community Plus - Customer Portal Manager

Category	Category API Value	User Licenses
		• Customer Portal Manager Custom • Customer Portal Manager Standard • Customer Portal Manager User • External Apps • External Apps Plus • External Identity • External Who • Gold Partner • Guest • Guest License • High Volume Customer Portal • High Volume Portal • Ideas Only Site • Internal Portal User • Partner • Partner Community • Silver Partner • Standard Partner
All External Customer and Partner Logins	`${communitiesLogin}`	• Customer Community Login • Customer Community Plus Login • External Apps Login • External Apps Plus Login • IdeasOnlyPortal • Limited Customer Portal Manager Custom • Limited Customer Portal Manager Standard • Overage Authenticated Website • Overage Customer Portal Manager Custom • Overage Customer Portal Manager Standard • Overage High Volume Customer Portal • Partner Community Login
External Customer Members	`${customerCommunities}`	• Authenticated Website • Chatter External • Company Communities

Category	Category API Value	User Licenses
		• Customer Community • Customer Community Plus • Customer Portal Manager • Customer Portal Manager Custom • Customer Portal Manager Standard • Customer Portal Manager User • External Apps • External Who • External Identity • Guest • Guest License • High Volume Customer Portal • High Volume Portal • Ideas Only Site • Internal Portal User
External Customer Logins	`${customerCommunitiesLogin}`	• Customer Community Login • Customer Community Plus Login • External Apps Login • IdeasOnlyPortal • Limited Customer Portal Manager Custom • Limited Customer Portal Manager Standard • Overage Authenticated Website • Overage Customer Portal Manager Custom • Overage Customer Portal Manager Standard • Overage High Volume Customer Portal
External Partner Members	`${partnerCommunity}`	• Bronze Partner • Channel Account • External Apps Plus • Gold Partner • Partner • Partner Community • Silver Partner • Standard Partner

Category	Category API Value	User Licenses
External Partner Logins	`${partnerCommunityLogin}`	- External Apps Plus Login - Partner Community Login

SEE ALSO:

Salesforce Help: User Licenses

Licensing Components (Developer Preview)

Create Custom Permission Sets (Developer Preview)

The permission set provides the grant to access the package features already entitled by a license.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Create your permission sets in a scratch org created from your new Dev Hub.

1. In your scratch org, from Setup, in the Quick Find Box, enter *`Permission Sets`*.
2. Select **Permission Sets**.
3. Click **New**.
4. Enter a label. The API Name field is generated automatically.
5. Enter a description.
6. For Select the type of users who will use this permission set, select **None**.

   Note: Packaging permission sets constrained to custom permission set licenses isn't currently supported.

7. Select the permissions to include in your permission set. When selecting the permissions, include your licensed custom permission.

SEE ALSO:

Salesforce Help: Permission Sets

Partner Licensing Platform Components and Concepts (Developer Preview)

Design Permission Sets (Developer Preview)

Create Custom Permission Set Groups (Developer Preview)

Permission set groups bundle permission sets together based on user job functions. Create permission set groups if your licensing structure requires a persona to have permissions from multiple custom permission set licenses.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Create your permission set groups in a scratch org created from your new Dev Hub.

1. In your scratch org, from Setup, in the Quick Find Box, enter `Permission Set Groups`.
2. Select **Permission Set Groups**.
3. Click **New**.
4. Enter a label. The API Name field is generated automatically.
5. Enter a description.
6. In the Permission Sets section, click **Permission Sets in Group**.
7. Click **Add Permission Set**.
8. Add all the permission sets related to the custom permission set license.

SEE ALSO:

Salesforce Help: Permission Set Groups

Partner Licensing Platform Components and Concepts (Developer Preview)

Design Permission Sets (Developer Preview)

Create Access Checks in Apex (Developer Preview)

Create access checks for your licensed custom permissions in your Apex code to gate access to your features.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn’t generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Create your access checks in a scratch org created from your new Dev Hub.

Use the FeatureManagementClass Apex class to gate your feature via the licensed custom permission. For example:

```
Boolean userHasCustomPermission =
FeatureManagement.checkPermission('{Licensed_Custom_Permission}');

if (userHasCustomPermission) {
 // your gated feature
}
```

For a more robust mechanism for doing licensing access checks, check out the LicensingUtil class in our demo application.

Note: Your new licensing can only be tested via an installed package, as a subscriber of your package. Testing your new licensing directly in a developer scratch org with deployed unmanaged metadata and code isn’t currently supported, since assignable and provisioned custom permission set license seats aren’t available in the developer org. For testing in the developer preview, the subscriber org where the managed package is installed is provisioned a default count of 10 seats for each custom permission set license in the package.

For this reason, the LicensingUtil class in our demo app bypasses the access check in certain developer orgs, while maintaining the access check in a subscriber context where the package is installed. This configuration allows all users to access the licensed functionality in those developer contexts as well as in automated Apex tests.

SEE ALSO:

Partner Licensing Platform Components and Concepts (Developer Preview)

Entitlements and Grants (Developer Preview)

Deploy Your Licensing Structure and Create Your Package (Developer Preview)

After you implement your new licenses and settings, you're ready to create a new package with this metadata.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

1. When creating your new package, make sure to include all necessary components, including your licensed custom permissions, custom permission set license definitions, permission sets, and permission set groups.
2. For 1GP, deploy the metadata to your packaging org and create a new managed package.
3. For 2GP, add the metadata source to your package directory's source and create a new managed package using SFDX.

Test Your Licensing Structure (Developer Preview)

After creating your new package, you can test your new licensing structure end-to-end.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Test your new licensing structure in a testing scratch org created from your new Dev Hub that is enabled with the Partner Licensing Platform. To enable the Partner Licensing Platform in this org, make sure that you have the `PartnerLicensingPlatform` feature in your scratch org definition file.

In this scratch org, install your newly created package to begin testing. Your new licensing can only be tested via an installed package, as a subscriber of your package. Testing your new licensing directly in a developer scratch org with deployed unmanaged metadata and code isn't currently supported, since assignable and provisioned custom permission set license seats aren't available in the developer org. For testing in the developer preview, the subscriber org where the managed package is installed is provisioned a default count of 10 seats for each custom permission set license in the package.

Best Practices for Testing Your Package Licensing (Developer Preview)

Verify your new licensing structure and whether your existing package works as expected with the new components.

Example: Test Plan for Validating License Design (Developer Preview)
Review this sample licensing test plan. You can adapt it to fit your package's needs.

SEE ALSO:

Salesforce Architects Blog: Find Bugs Earlier with Second-Generation Packaging

Best Practices for Testing Your Package Licensing (Developer Preview)

Verify your new licensing structure and whether your existing package works as expected with the new components.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

At minimum, we recommend that you test these categories.

- Perform regression tests to make sure product functionality is working as before.
- Manually validate your licensing structure and ensure your new licenses and permission sets are authorizing access to features appropriately.
  - Check for under-permissioning, or if users have less access than expected when assigned permission sets and custom permission set licenses.
  - Check for over-permissioning, or if users have more access than expected when assigned permission sets and custom permission set licenses.
- Make sure that users can only be assigned the intended custom permission set licenses as specified using user license restriction categories. For example, users with licenses in the External Customer Logins category user can only be assigned the license meant for them.

Since testing your new licensing directly in a developer scratch org isn't currently supported, the LicensingUtil class in our demo app bypasses the access check in certain developer orgs, while maintaining the access check in a subscriber context where the package is installed. This configuration allows all users to access the licensed functionality in those developer contexts.

Your current automated Apex tests may begin to fail with the new licensing access checks and may need refactoring. Currently, automated testing of licensed functionality via license and permission set assignment to a user isn't supported, and may only work on the subscriber side where the package is installed. You may need to bypass your new access checks during automated testing, as shown in the LicensingUtil class example.

Note: Scratch orgs always operate with "site-wide" package licensing. As a result, all users in your subscriber scratch org are entitled to your foundation package features, such as Visualforce pages, without a managed package license or a custom (foundation) permission set license. Real subscribers of your package will need either a managed package license or custom permission set license to access your package's foundation features.

Example: Test Plan for Validating License Design (Developer Preview)

Review this sample licensing test plan. You can adapt it to fit your package's needs.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Our forward-looking statement applies to custom supplement permission set licenses. Make your purchasing decisions based on currently available technology.

This test plan continues the Travel Navigation example from the Design Your Package Licensing Structure section of this guide.

Package Access

Testing Scenario	Expected Behavior
User is assigned Territory Planning CPSL	User can’t access Travel Navigation package
User is assigned Maps CPSL	User can access Travel Navigation package
User is assigned Maps Advanced CPSL	User can access Travel Navigation package
User is assigned Territory Planning CPSL and Maps CPSL	User can access Travel Navigation package

Feature Access

Testing Scenario	Expected Behavior
User is assigned: • Maps CPSL • Territory Planning CPSL • Service Agent CPSL • Service Manager CPSL • Org Manager permission set group	User can access: • Travel Navigation package • Territory planning • Triaging service requests • Escalating service requests
User is assigned: • Maps CPSL • Territory Planning CPSL • Service Agent CPSL • Service Manager CPSL User isn't assigned: • Org Manager permission set group	User can access: • Travel Navigation package User can't access: • Territory planning • Triaging service requests • Escalating service requests
User is assigned: • Maps CPSL • Territory Planning CPSL • Territory Planner permission set	User can access: • Travel Navigation package • Territory planning User can't access: • Triaging service requests • Escalating service requests

Testing Scenario	Expected Behavior
User is assigned: • Maps CPSL • Territory Planning CPSL User isn't assigned: • Territory Planner permission set	User can access: • Travel Navigation package User can't access: • Territory planning • Triaging service requests • Escalating service requests
User is assigned: • Maps CPSL • Territory Planner permission set User isn't assigned: • Territory Planning CPSL	Territory Planner permission set assignment fails, because Territory Planning CPSL isn't assigned. User can access: • Travel Navigation package

Permission Set License Assignments

Testing Scenario	Expected Behavior
Internal user is assigned Maps Community PSL	Error when assigning permission set license
External user is assigned Maps CPSL	Error when assigning permission set license

Manage Migrations to the Partner Licensing Platform (Developer Preview)

If you have existing customers for your product, you can take steps to ensure that both your new and existing license settings work while you transition to your new licensing design.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

When you're ready, communicate the changes to your licensing structure and your timeline for migrating to the new licensing design to your existing customers.

Manage a Hybrid Licensing Model During Migrations (Developer Preview)

Set up your access checks so that your new custom permission set licenses work alongside your existing managed package licenses, and both new and existing customers can use your features without disruption.

Licensing for Existing Custom Permissions (Developer Preview)

Set up your existing custom permissions and new licensed custom permissions so that both new and existing customers can access your features as you migrate to your new licensing design.

Package Access Validation in a Hybrid Licensing Model (Developer Preview)

After a subscriber's org is provisioned custom permission set licenses, going forward, package access for the namespace is validated whenever permission sets are assigned to users.

Manage a Hybrid Licensing Model During Migrations (Developer Preview)

Set up your access checks so that your new custom permission set licenses work alongside your existing managed package licenses, and both new and existing customers can use your features without disruption.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

New customers will use your new licensing structure and settings from the beginning. However, if you upgrade your package code with your new access checks and settings, this change can cause your existing customers to lose access to those newly gated existing features, because users may not have the necessary licenses and permissions assigned for your new license settings at the time of upgrade. By setting up conditional behavior, both new and existing customers can access features without disruption.

Wherever you have access checks for your new licensed custom permissions, add an OR condition that also checks whether the user has the original managed package license for the package. Use the hasPackageLicense method, which returns true if the context user has a license to the managed package via a package license only.

For example:

```
Boolean hasCustomPermission =
FeatureManagement.checkPermission('{Licensed_Custom_Permission}');
if (hasCustomPermission ||
UserInfo.hasPackageLicense([Your_Package_Id]) {
//your gated feature}
```

Note: For a more robust mechanism for doing licensing access checks, including this hybrid licensing model, check out the LicensingUtil class in our demo application. However, because scratch orgs always operate with "site-wide" package licensing, the hasPackageLicense method always returns `true` in this context. Currently, this hybrid licensing access check will only be applicable for real subscribers of your package, who will need either a managed package license or custom (foundation) permission set license to access your package.

To break down how both new and existing customers interact with this access check:

- All new users or customers start with your new licenses. If they have the appropriate licensed custom permission, they pass the access check.
- All existing customers' users with managed package licenses retain their original access by passing the hasPackageLicense access check.
- As users migrate onto your new licenses, their original managed package licenses should also be removed. At this point, users can only access the feature if they have the appropriate licensed custom permission.

With this approach, your customers have time to understand and migrate onto your new licenses without losing access.

Licensing for Existing Custom Permissions (Developer Preview)

Set up your existing custom permissions and new licensed custom permissions so that both new and existing customers can access your features as you migrate to your new licensing design.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

After a licensed custom permission is packaged and released, you can't edit the **License Required** attribute, which means that existing custom permissions can't be converted to a licensed custom permission. This behavior is because users who already have the custom permission may not have the necessary license assigned when the custom permission becomes licensed at the time of upgrade.

Because you can't directly convert existing custom permissions to licensed ones, set up your access checks to support both existing custom permissions and new licensed custom permissions. First, create your licensed custom permission. Then, locate in your access checks where you currently check for the original custom permission. Instead of replacing the original custom permission with your new licensed custom permission, put an `OR` clause between the two custom permissions. As a result, existing customers and users can continue to get uninterrupted access to the gated feature, although without a license. New customers and users get access to the gated feature via the new licensed custom permission and new license.

After you communicate the change and migration expectation to your existing customers and give them time to adopt the new licensed custom permission, you can remove the old custom permission from your Apex access checks to enforce the migration and license assignment.

Package Access Validation in a Hybrid Licensing Model (Developer Preview)

After a subscriber's org is provisioned custom permission set licenses, going forward, package access for the namespace is validated whenever permission sets are assigned to users.

Note: The Partner Licensing Platform is available as a developer preview. The Partner Licensing Platform isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in your production package.

Our forward-looking statement applies to partner business org provisioning. Make your purchasing decisions based on currently available technology.

When an admin assigns a user a permission set that grants access to package components, the system first validates whether the user has a custom foundation permission set license, or managed package license, that allows access to the package namespace. Previously, package access wasn't validated until the user attempted to access the components. This delay can cause an inconsistent authorization experience, because admins can assign users permission sets that grant access to components in a package, while those users may not have access to the package itself via a license.

For your existing customers, users with managed package licenses can continue to access package components via a permission set assignment. However, for the users who were assigned permission sets that grant access to package components–but who don't have a managed package license or custom foundation permission set license assigned–this validation will now trigger an error. This behavior makes for a consistent authorization experience across both Salesforce and ISV products and licenses. The subscriber admin must either remove the permission set assignment, assign the user a custom permission set license, or remove access to the managed package components from the permission set.