

# AppExchange Packaging Guide



Last updated: July 6, 2009



# Table of Contents

About Force.com AppExchange Packaging.....	2
Quick Start Guide.....	2
Packaging Process.....	3
Force.com AppExchange Basics.....	5
About Force.com AppExchange.....	5
Packaging Glossary.....	6
About Apps.....	9
Who Can Use Force.com AppExchange?.....	11
Preparing Your App for Force.com AppExchange.....	12
Deciding What to Package.....	12
Branding Considerations.....	12
Understanding Composite, Native, and Client Apps.....	13
Charging for Use of Your App.....	13
Component Behavior in Packages.....	13
Publishing Your Apps on Force.com AppExchange.....	20
Protecting Your Intellectual Property.....	21
Designing Custom Profile Settings.....	21
Tips for Building Your App.....	23
Developing App Documentation.....	23
Managing License Agreements.....	25
Working with External Services.....	25
Provisioning External Services.....	25
Uninstalling Apps.....	26
Matching the Salesforce User Interface.....	26
Packaging and Uploading Apps.....	27
Preparing Your Apps for Distribution.....	27
Publishing and Listing Your App.....	31
Supporting Apps.....	32
Providing End User Support.....	32
Promoting Your App.....	32
Upgrading Packages.....	33
Monitoring Usage of Your App.....	34
Billing for Use of Your App.....	34
Checklist for Success.....	34
Frequently Asked Questions.....	34
Index.....	35

# About Force.com AppExchange Packaging

Force.com AppExchange is an application-sharing service from salesforce.com that allows you to browse, install, and share apps and components stored in packages and built for the Force.com platform. Custom apps can contain custom objects, custom tabs, reports, and many other items that are packaged together and published on Force.com AppExchange.

This guide is designed for Salesforce users and partners who want to package and upload their app to AppExchange. Use this guide to learn how to package apps for AppExchange, including tips and best practices for preparing, building, and packaging your app.

See [Quick Reference for Developing Packages](#) for quick reference information that is useful when you are starting to design and develop packages. It contains tables describing the components that can be included in packages, and which component attributes can be edited in managed packages. It also shows how some component behaviors change depending on how they are packaged.

## Quick Start Guide

Use this quick start checklist to create and package your app for Force.com AppExchange. Click on any topic below to get more detail.

### Review the basic concepts

- Find out what it is and how it works
- Get a list of what components you can include in an app
- Review AppExchange terminology

### Prepare your app and its components

- Determine the components you want to package
- Decide your branding, the type of app you want to develop, and whether you want to charge for your app
- Know what to customize before making your app available on Force.com AppExchange

### Build your app

- Use the Salesforce online help to learn how to create custom objects, tabs, or any other component you want included in your app
- Develop documentation for users of your app as well as administrators that install it
- Consider handling external services, license agreements, and how users will uninstall your app
- Find out about matching the Salesforce user interface

### Package and upload your app

- Add components to your package
- Upload your app

### Publish and post your app

- Go to [sites.force.com/appexchange/publisherHome](https://sites.force.com/appexchange/publisherHome) for information on creating your listing, assigning a publishing profile, getting your app inspected, and posting it

### **Support your app**

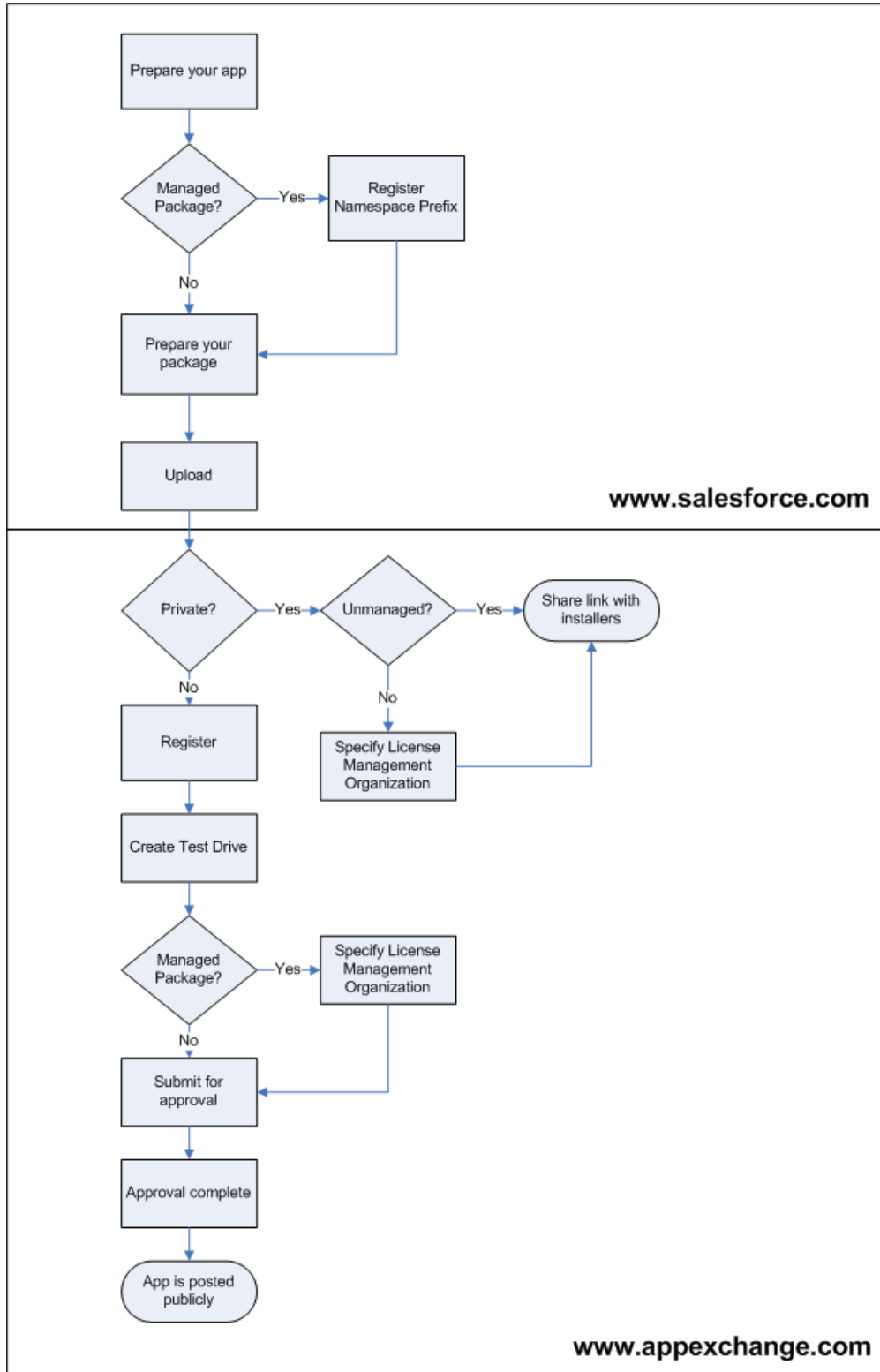
Learn how to be prepared for users who may need support

Find out about locking down your app, monitoring usage, and promoting your app

Review a checklist for success and find out where to get answers to frequently asked questions

## **Packaging Process**

Before you begin, review the high-level process for packaging your app and making it available on Force.com AppExchange:



# Force.com AppExchange Basics

Force.com AppExchange gives Salesforce users a new venue to share pre-built customizable apps. To get started packaging your apps for AppExchange, review these basics:

## About Force.com AppExchange

Discover what it is and how it works.

## Packaging Glossary

Review the terminology used in this document and on AppExchange.

## About Apps

Find out what types of components you can include in your app.

## Who Can Use Force.com AppExchange?

Determine what users can use AppExchange.

## About Force.com AppExchange

Available for: **Enterprise, Unlimited, Professional, Developer, and Group** Editions

User Permissions Needed	
To install or uninstall packages from AppExchange	"Download AppExchange Packages"
To create new AppExchange packages	"Create AppExchange Packages"
To create test drives and publish AppExchange packages to AppExchange	"Upload AppExchange Packages"

Force.com AppExchange is an application-sharing service from salesforce.com that allows you to browse, install, and share apps and components stored in packages and built for the Force.com platform. Each app is uploaded to Force.com AppExchange, where it can be listed in appropriate categories to make it easy for users to find the app they need. Users can share uploaded apps privately with a particular user or publicly with all users on AppExchange.

AppExchange supports three types of apps:

- A native app is built 100% on the Force.com platform.
- A composite app contains elements from the Force.com platform and other elements from external systems.
- A client app accesses Salesforce via the Web Services API, but does not use any other part of the Force.com platform.

The AppExchange supports the following roles throughout the life of an app:

### A developer creates an app

The developer of an app is the Salesforce user who creates an app. If an app is a package, it may contain an app and its components, including custom objects, custom tabs, reports, and many other items. All apps integrate with Salesforce.

### A publisher publishes the app

The publisher of an app is the Salesforce user or organization who creates a profile and one or more listings for the app on AppExchange. Publishers listings contain a link to an app they have uploaded to AppExchange, or to a third-party website. Publishers also set default license settings for managed packages.

### An installer installs the app in a Salesforce organization

The installer of a package is the person who visits the AppExchange, finds a package, and installs it into a Salesforce organization.

### Users use the app

Once the publisher installs the app from the AppExchange into a Salesforce organization, users in that organization can use the app.

After users install an app, they can customize it to fit the needs of their business. Therefore, the same installed app may have different custom fields, tabs, or custom s-controls in other Salesforce organizations.

The same person may perform one or more roles.

Like most Internet marketplaces, AppExchange contains detailed information about each app:

- Anyone can search for apps, and explore detailed specifications, reviews of the app from the Salesforce user community, or study ratings of the app. The publisher of the app is listed, so anyone can contact the publisher with questions.
- Users, who may also be installers or developers, can also:
  - Post reviews.
  - Click **Test Drive** to run a fully-functioning demo of the app as a read-only user in a familiar Salesforce environment.
  - Click **Get It Now** to begin the install process.
- Publishers can create public and private listings for their apps, and create or modify a publisher profile.



**Note:** All apps publicly listed on AppExchange must pass the salesforce.com [security review](#).

## Packaging Glossary

Available in: **Group, Professional, Enterprise, Unlimited, and Developer** Editions

Review the following terminology with Force.com AppExchange:

### App

A collection of components such as tabs, reports, dashboards, and Visualforce pages that address a specific business need. Short for "application." Salesforce provides standard apps such as Sales and Call Center. You can customize the standard apps to match the way you work. In addition, you can package an app and upload it to Force.com AppExchange along with related components such as custom fields, custom tabs, and custom objects. Then, you can make the app available to other Salesforce users from AppExchange.



## AppExchange Directory

A Web directory where hundreds of apps are available to Salesforce customers to review, demo, comment upon, and/or install. Developers can submit their apps for listing on [AppExchange](#) if they wish to share them with the community.

## Beta, Managed Package

In the context of managed packages, an early version of a managed package that is uploaded to Force.com AppExchange for a sampling of your intended audience to test it.

## Deploy

To move functionality from an inactive state to active. For example, when developing new features in the Salesforce user interface, you must select the "Deployed" option to make the functionality visible to other users.

The process by which an application or other functionality is moved from development to production.

To move metadata components from a local file system to a Salesforce organization.

For installed apps, deployment makes any custom objects in the app available to users in your organization. Before a custom object is deployed, it is only available to administrators and any users with the "Customize Application" permission.

## Deprecated Component

A publisher may decide to refine the functionality in a managed package over time as the requirements evolve. This may involve redesigning some of the custom objects or fields in the managed package. Publishers cannot delete some components in a Managed - Released package, but they can deprecate a component in a later package version so that new subscribers do not receive the component, while the component continues to function for existing subscribers and API integrations. A publisher can deprecate Apex methods, classes, exceptions, enums, interfaces, or variables.

## License Management Application (LMA)

A free AppExchange app that allows you to track sales leads and accounts for every user who downloads a managed package of yours from AppExchange.

## License Management Organization (LMO)

The Salesforce organization that you use to track all the Salesforce users who install your package. A license management organization must have the License Management Application (LMA) installed. It automatically receives notification every time your package is installed or uninstalled so that you can easily notify users of upgrades. You can specify any Enterprise, Unlimited, or Developer Edition organization as your license management organization. For more information, go to [sites.force.com/appexchange/publisherHome](https://sites.force.com/appexchange/publisherHome).

## Managed Package

A collection of application components that are posted as a unit on AppExchange, and are associated with a namespace and a License Management Organization. A package must be managed for it to be published publicly on AppExchange, and for it to support upgrades. An organization can create a single managed package that can be downloaded and installed by many different organizations. They differ from unmanaged packages in that some components are locked, allowing the managed package to be upgraded later. Unmanaged packages do not include locked components and cannot be upgraded. In addition, managed packages obfuscate certain components (like Apex) on subscribing organizations, so as to protect the intellectual property of the developer.

## Managed Package Extension

Any package, component, or set of components that adds to the functionality of a managed package. An extension requires that the base managed package be installed in the organization.

## Namespace Prefix

In a packaging context, a namespace prefix is a one to 15-character alphanumeric identifier that distinguishes your package and its contents from packages of other developers on AppExchange. Namespace prefixes are case-insensitive. For example, ABC and abc are not recognized as unique. Your namespace prefix must be globally unique across all Salesforce organizations. It keeps your managed package under your control exclusively.

## Native App

An app that is built exclusively with setup (metadata) configuration on Force.com. Native apps do not require any external services or infrastructure.

## Package

A group of Force.com components and applications that are made available to other organizations through the AppExchange. You use packages to bundle an app along with any related components so that you can upload them to Force.com AppExchange together.

## Package Dependency

Created when one component references another component, permission, or preference, which must exist for the component to be valid. Components can include but are not limited to:

- Standard or custom fields
- Standard or custom objects
- Visualforce pages
- Apex scripts

Permissions and preferences can include but are not limited to:

- Divisions
- Multicurrency
- Record types

## Package Installation

Incorporates the contents of a package into your Salesforce organization. A package on AppExchange can include an app, a component, or a combination of the two. After you install a package, you may need to deploy components in the package to make it generally available to the users in your organization.

## Package Publication

Publishing your package makes it publicly available on Force.com AppExchange. Apps can be found under specific categories and by doing a search for keywords.

## Package Registration

Registering your package allows you to access information about users who visit or download your app from the Force.com AppExchange.

## Package Version

A package version is a set of components uploaded in a package. It is denoted by a version number, *majorNumber.minorNumber*, such as 2.1. Unmanaged packages are not upgradeable so each package version is simply a set of components for distribution. A package version has more significance for managed packages. Packages can exhibit different behavior for different versions. Publishers can use package versions to evolve the components in their managed

packages gracefully by releasing subsequent package versions without breaking existing customer integrations using the package.

### Private Sharing

The process of privately sharing a package uploaded to the AppExchange, by using the URL that you receive from Salesforce upon upload. This URL is not listed in the AppExchange. Using the unlisted URL allows you to share a package manually without making it public.

### Publisher

The publisher of a package is the Salesforce user or organization that published the package on AppExchange. Publishers are required to register using their contact information after registering their package on AppExchange.

### Subscriber

The subscriber of a package is a Salesforce user with an installed package in their Salesforce organization.

### Test Drive

A test drive, also known as a demo, is a fully functional Salesforce organization that contains an app and any demo records added by the publisher for a particular package. It allows users on Force.com AppExchange to experience an app as a read-only user using a familiar Salesforce interface.

### Unmanaged Package

An AppExchange package that cannot be upgraded or controlled by its developer. Unmanaged packages allow you to take any app components and move them "as is" to AppExchange without going through a lengthy publishing process.

### Upgrading

Upgrading a package is the process of installing a newer version. Salesforce supports upgrades for managed packages that are not beta.

### Uploading

Uploading a package sends it to Force.com AppExchange. Uploading your package is just the first step; all apps on AppExchange must be registered, making them privately available. The final and optional step is to publish it, making it publicly available on AppExchange.

## About Apps

Available in: **Group, Professional, Enterprise, Unlimited, and Developer** Edition

An *app* is a collection of components such as tabs, reports, dashboards, and Visualforce pages that address a specific business need. Short for "application." Salesforce provides standard apps such as Sales and Call Center. You can customize the standard apps to match the way you work. In addition, you can package an app and upload it to Force.com AppExchange along with related components such as custom fields, custom tabs, and custom objects. Then, you can make the app available to other Salesforce users from AppExchange. An app can contain one or more of the following components:

- Apex classes or triggers (available in Developer, Enterprise, and Unlimited Editions only)
- Apex sharing reason
- Apex sharing recalculation
- Custom apps

- Custom fields on standard objects
- Custom buttons and links
- Custom home page layouts (including their custom components)
- Custom labels
- Custom objects (including all custom fields, relationships (master-detail and lookup), picklist values, record types, page layouts, mini page layouts, related list layouts, search layouts, public list views, validation rules, custom buttons, custom links, its Context-Sensitive Help Setting, and the Custom S-Control selected for it)



**Note:** When packaged and installed, only public list views from an app are installed. If a custom object has any custom list views that you want to include in your package, ensure that the list view is accessible by all users.

- Custom report types
- Custom reports
- Custom s-controls
- Custom tabs
- Dashboards
- Documents
- Email templates (including any letterheads that they use)
- Folders
- Letterheads
- List views
- Page layouts
- Profiles
- S-Controls
- Static resources
- Record types
- Reports
- Translated values available in the translation workbench for packaged components
- Visualforce pages
- Workflow rules, alerts, field updates, outbound messages, and tasks
- Validation rules
- Visualforce components
- Visualforce pages

When adding components to a package, some related components are automatically included. For example, when you add a custom object to a package, its custom fields, page layouts, and relationships with standard objects are automatically included. Some components are not visible in the list of packaged components; for example, translated values for custom fields are included for any active language that has been added to the package yet the custom field is displayed in the developer's language on the package detail page.



**Note:** The custom apps, tabs, and objects contained in a managed package in a trial state or the managed package of a native application in any state do not count against the limits supported for your Salesforce Edition.

## Who Can Use Force.com AppExchange?

Anyone with a valid Salesforce username and password, including trial users, can use AppExchange to browse and test drive apps. Salesforce administrators and users with the "Customize Application" permission can install and deploy AppExchange packages into their own environments. To allow other users in your organization to use AppExchange, give them the appropriate user permission listed below:

Permission Name	Description
"Create AppExchange packages"	Create new AppExchange packages
"Download AppExchange packages"	Install or uninstall packages from AppExchange
"Upload AppExchange packages"	Create test drives and upload AppExchange packages to AppExchange

# Preparing Your App for Force.com AppExchange

Design your app carefully with Force.com AppExchange in mind. Use the following suggestions to determine what components to include and how to package them:

## Decide what to publish

Decide what components are included in your app.

## Consider branding for your app

Evaluate how you want to brand your app.

## Review the differences between composite, native, and client apps

Determine what type of app you want to develop.

## Determine if you want to charge for your app

Review your options for charging others to use your app.

## Design your app for Force.com AppExchange

Customize any custom objects, tabs, folders, reports, dashboards, or page layouts for distribution on AppExchange.

## Publish your app on Force.com AppExchange

Determine how you want to publish your app on AppExchange.

## Review how to protect your intellectual property

Determine if you need to hide any complex customizations within your app.

## Configure your custom profiles for Force.com AppExchange

Give your custom profiles meaningful names and descriptions so they are easy to identify when installed in another Salesforce organization.

## Deciding What to Package

Packaging for Force.com AppExchange is flexible. You can package a complete app, a single component (such as a report), or a collection of components..

Listing your app on AppExchange is also flexible. You are not limited to a single listing. You may post one app that addresses a generic business need and also post several variations of the app that are specific to certain vertical markets.

To determine what you would like to package, consult Salesforce IdeaExchange at [ideas.salesforce.com](https://ideas.salesforce.com) for the features our customers demand.

## Branding Considerations

Consider the importance of branding while designing your app. Determine a meaningful and powerful brand that you can use consistently on AppExchange to identify your app. See *Force.com AppExchange Partner Logo and Naming Guidelines* on [sites.force.com/appexchange](https://sites.force.com/appexchange) for app logo and naming conventions.

## Understanding Composite, Native, and Client Apps

Before designing your app for AppExchange, review the different types of apps:

### Native apps

Native apps are built using only the resources available on the Force.com platform. The platform includes a powerful security model that you can use to control the visibility of all information down to the field level. With the power of formula fields, custom links, and custom s-controls, you can automate your business process without the need for external services.

### Composite apps

Composite apps are built with a combination of the Force.com platform and external services. The external services may be publicly available and free, such as Google maps or FedEx package tracking services, or they may be services that you create and sell. See [Charging for Use of Your App](#) on page 13.

### Client apps

Client apps are built using only the Force.com API to access the Salesforce service. They do not alter or customize the Salesforce user interface. For example, Connect for Outlook transfers emails directly from Microsoft® Outlook® into the Salesforce service without ever leaving the Outlook application on your client.

## Charging for Use of Your App

You can choose if you want to charge for your app or use an AppExchange listing to drive interest to other business services that you provide. AppExchange does not currently provide a billing mechanism. However, you can build an app in Salesforce that links to an external service through a custom s-control, web tab, or other component. Then, you can derive a revenue stream by charging an installation or subscription fee for your external service. The exchange of funds for this service must occur outside of Salesforce. If you choose to do this, we recommend storing your pricing and licensing information in a file within the Documents tab of your app. Then, point users to this document from the About tab of your app.

## Component Behavior in Packages

Available in: <b>Group, Professional, Enterprise, Unlimited,</b> and <b>Developer</b> Editions
--

User Permissions Needed	
To create AppExchange packages:	"Create AppExchange Packages"

When building an app for distribution, it is important to consider how packaging affects your app and its components. This section contains guidelines and information on special behaviors in packages. Use these guidelines to help you determine what to include in your packages, how to design your app, and how to distribute your packages (managed or unmanaged).

**Note:**

- For more information on the properties of each component in packages, see the "packaged components properties table" in the Salesforce online help. For more information on the attributes of each component in packages, see the "component attributes table" in the Salesforce online help.
- Component names must be unique within an organization. To ensure that your component names do not conflict with those in an installer's organization, use a managed package so that all of your component names contain your namespace prefix.

**Analytic Snapshot**

Developers of managed packages must consider the implications of introducing analytic snapshots that reference reports released in a previous version of the package. If the subscriber deleted the report or moved the report to a personal folder, the analytic snapshot referencing the report will not be installed, even though the Package Installation page may indicate that it will be. Also, if the subscriber has modified the report, that report may return results impacting the information displayed by the analytic snapshot. As a best practice, the developer should release an analytic snapshot and the related reports in the same version.

Custom objects are developer-controlled so analytic snapshot field mappings could become invalid when the package is upgraded. For example, if the developer deletes a custom field included in the field mapping, it would become invalid. Analytic snapshots can be created, installed, and run with invalid mappings as long as one valid mapping exists.

Since the running user is selected by the subscriber, some analytic snapshot field mappings could become invalid if the running user does not have access to source or target fields.

**Apex Classes or Triggers**

Any Apex that is included as part of a Force.com AppExchange package must have at least 75% cumulative test coverage. Each trigger must also have some test coverage. When you upload your package to AppExchange, all tests are run to ensure that they run without errors. In addition, all tests are run when the package is installed in the installer's organization. The installer can decide whether or not to install the package if any tests fail.



**Tip:** To prevent naming conflicts, Salesforce recommends using managed packages for all packages that contain Apex. This way, all of the Apex objects contain your namespace prefix. For example, if there is an Apex class called `MyHelloWorld` and the namespace for your organization is `OneTruCode`, the class is referenced as `OneTruCode:MyHelloWorld`.

Keep the following considerations in mind when including Apex in your package:

- If you are exposing any methods as Web services, include detailed documentation so that subscribers can write external code that calls your Web service.
- If an Apex class references a custom label, and that label has translations, you must explicitly package the individual languages desired in order for those translations to be included in the package.
- If you reference a custom object's sharing object (such as `MyCustomObject__share`) in Apex, this adds a sharing model dependency to your package. You must set the organization-wide sharing default access level for the custom object to Private in order for other organizations to install your package successfully. For more information on setting the sharing model, see "Managing the Sharing Settings" in the Salesforce online help.
- The code contained in an Apex script that is part of a managed package is automatically obfuscated and cannot be viewed in an installing organization. The only exceptions are methods declared as global, meaning that the method signatures can be viewed in an installing organization.
- You can use the `deprecated` annotation in Apex to identify methods, classes, exceptions, enums, interfaces, and variables that can no longer be referenced in subsequent releases of the managed package in which they reside. This



is useful when you are refactoring code in managed packages as the requirements evolve. After you upload another package version as Managed - Released, new subscribers that install the latest package version cannot see the deprecated elements, while the elements continue to function for existing subscribers and API integrations.

### Custom Fields

- Picklist field values for custom fields can be added, edited, or deleted by subscribers. A developer should carefully consider this when explicitly referencing a picklist value in code. Picklist values can be added or deleted by the developer. During a package upgrade, no new picklist values are installed into the subscriber's organization for existing fields. Any picklist values deleted by the developer are still available in the subscriber's organization.
- Auto-number type fields and required fields cannot be added after the object is uploaded in a managed-released package.
- Developers can add universally required custom fields to managed packages as long as they have default values.

### Custom Labels

If a label is translated, the language must be explicitly included in the package in order for the translations to be included in the package. Subscribers can override the default translation for a custom label.

### Custom Objects

- If a developer enables the `Allow Reports` or `Allow Activities` attributes on a packaged custom object, the subscriber's organization also has these features enabled during an upgrade. Once enabled in a managed-released package, the developer and the subscriber cannot disable these attributes.
- Standard button and link overrides are not packageable. The subscriber must manually configure these settings after installation.

### Custom Report Types

A developer can edit a custom report type in a managed package after it is released. Subscribers automatically receive these changes when they install a new version of the managed package. However, developers cannot remove objects or fields from the report type once the package is released.

### Custom Tabs

- The `Tab Style` for a custom tab must be unique within your app. However, it does not need to be unique within the organization where it is installed. A custom tab's style will not conflict with an existing custom tab in the installer's environment.
- If you want to provide custom tab names in different languages, see "Renaming Tab and Field Labels" in the Salesforce online help. Subscribers cannot edit custom tabs in a managed package.

### Customer Portal and Partner Portal

Packages referring to Customer Portal or Partner Portal fields are supported. The subscriber installing the package must have the respective portal enabled to install the package.

### Dashboard Components

Developers of managed packages must consider the implications of introducing dashboard components that reference reports released in a previous version of the package. If the subscriber deleted the report or moved the report to a personal folder, the dashboard component referencing the report is dropped during install. Also, if the subscriber has modified

the report, that report may return results impacting what information is displayed by the dashboard component. As a best practice, the developer should release a dashboard and the related reports in the same version.

### Divisions

- When divisions are enabled on a custom object in a package, the subscribing organization must have the divisions feature enabled to install the package.
- Setting the division filter on a report does not cause a dependency. The setting is dropped when installed into the subscriber's organization.
- Summarizing by the object's division field—for example, Account Division—in a report causes a dependency.
- If the object's division field in a report is included as a column, and the subscriber's organization does not support divisions on the object, then the column is dropped during install.
- If you install a custom report type that includes an object's division field as a column, that column is dropped if the organization does not support divisions.

### Field Dependencies

- Developers and subscribers can add, change, or remove field dependencies.
- If the developer adds a field dependency, it is added during installation unless the subscriber has already specified a dependency for the same field.
- If a developer removes a dependency, this change is not reflected in the subscriber's organization during an upgrade.
- If the developer introduces a new picklist value mapping between the dependent and controlling fields, the mapping is added during an upgrade.
- If a developer removes a picklist value mapping, the change is not reflected in the subscriber's organization during an upgrade.

### Folders

- Components that Salesforce stores in folders, such as documents, cannot be added to packages when stored in personal and unfiled folders. Put documents, reports, and other components that Salesforce stores in folders in one of your publicly accessible folders.
- Components such as documents, email templates, reports, or dashboards are stored in new folders in the installer's organization using the publisher's folder names. Give these folders names that indicate they are part of the package.
- If a new report, dashboard, document, or email template is installed during an upgrade, and the folder containing the component was deleted by the subscriber, the folder is re-created. Any components in the folder that were previously deleted are not restored.
- The name of a component contained in a folder must be unique across all folders of the same component type, excluding personal folders. Components contained in a personal folder must be unique within the personal folder only.

### Home Page Components

When you package a custom home page layout, all the custom home page components included on the page layout are automatically added. Standard components such as Messages & Alerts are not included in the package and do not overwrite the installer's Messages & Alerts. To include a message in your custom home page layout, create an HTML Area type custom Home tab component that contains your message and add it to your custom home page layout.

## Home Page Layouts

Once installed, your custom home page layouts are listed with all the subscriber's home page layouts. Distinguish them by including the name of your app in the page layout name.

## List Views

List views associated with queues cannot be included in a package.

## Multi-Currency

- If a subscriber installs a report or custom report type that includes an object's currency field as a column, that column is dropped if the subscriber's organization is not enabled for multiple currencies.
- Referencing an object's currency field in a report's criteria—for example, `Account Currency`—causes a dependency.
- Summarizing by an object's currency field in a report causes a dependency.
- Using a currency designation in a report criteria value—for example, "Annual Revenue equals GBP 100"—does not cause a dependency. The report generates an error when run in the installers organization if it does not support the currency.
- If an object's currency field in a report is included as a column and the subscriber's organization is not enabled for multiple currencies, that column is dropped during install.
- If a subscriber installs a custom report type that includes an object's currency field as a column, that column is dropped if the organization is not enabled for multiple currencies.

## Page Layouts

The page layout of the person uploading a package is the layout used for Group and Professional Edition organizations and becomes the default page layout for Enterprise, Unlimited, and Developer Edition organizations.

## Person Accounts

Packages referring to fields related to person accounts on the Account object—for example, `Is Person Account`, `First Name`, `Last Name`, `Title`—cannot be uploaded.

## Picklist Values

- Subscribers can rename or delete picklist field values. A developer should carefully consider this when explicitly referencing a picklist field value in Apex.
- Picklist field values can be added or deleted in the developer's organization. Upon upgrade, no new values are installed. Any values deleted by the developer are still available in the subscriber's organization until the subscriber deletes them.

## Profile Settings

Once a managed package is  Managed - Released, the developer can make changes to the profile settings but subscribers will not get these changes when upgrading. Salesforce recommends contacting subscribers to ask them to make important profile changes manually when they upgrade.

## Record Types

- If record types are included in the package, the subscriber's organization must support record types to install the package.
- When a new picklist value is installed, it is associated with all installed record types according to the mappings specified by the developer. A subscriber can change this association.

- Referencing an object's record type field in a report's criteria—for example, `Account Record Type`—causes a dependency.
- Summarizing by an object's record type field in a report's criteria—for example, `Account Record Type`—causes a dependency.
- If an object's record type field is included as a column in a report, and the subscriber's organization is not using record types on the object or does not support record types, then the column is dropped during install.
- If you install a custom report type that includes an object's record type field as a column, that column is dropped if the organization does not support record types or the object does not have any record types defined.
- Record types on standard objects cannot be packaged and therefore should not be referenced by any components included in the package.

## Reports

If a report includes elements that cannot be packaged, those elements will be dropped or the package upload will fail. For example:

- Hierarchy drill-downs are dropped from activity and opportunities reports.
- Filters on unpackageable fields are automatically dropped (for example, in filters on standard object record types).
- Package upload fails if a report includes advanced filter conditions on an unpackageable field (for example, in filters on standard object record types).
- Lookup values on the `Select Campaign` field of standard campaign reports are dropped.
- Reports are dropped from packages if they have been moved to a private folder or to the Unfiled Public Reports folder.

## Translation Workbench

- If you have enabled the translation workbench and added a language to your package, any associated translated values are automatically packaged for the appropriate components in your package. Make sure that you have provided translations for all possible components.
- An installer of your package can see which languages are supported on the package detail page. The installer does not need to enable anything for the packaged language translations to appear. The only reason installers may want to enable the translation workbench is to change translations for unmanaged components after installation, override custom label translations in a managed package, or to translate into additional languages.
- If you are designing a package extension, you can include translations for the extension components but not additional translations for components in the base package.
- On install, a subscriber must activate a language if it is not active in the subscriber's organization.

## Workflow

- Salesforce prevents you from uploading workflow alerts that have a public group, partner user, or role recipient. Change the recipient to a user before uploading your app. During installation, Salesforce replaces that user with the user installing the app, and the installer can customize it as necessary.
- Salesforce prevents you from uploading workflow field updates that change an `Owner` field to a queue. Change the updated field value to a user before uploading your app. During installation, Salesforce replaces that user with the user installing the app, and the installer can customize it as necessary.
- Salesforce prevents you from uploading workflow rules, field updates, and outbound messages that reference a record type on a standard or managed-installed object.

- Salesforce prevents you from uploading workflow tasks that are assigned to a role. Change the `Assigned To` field to a user before uploading your app. During installation, Salesforce replaces that user with the user installing the app, and the installer can customize it as necessary.
- You can package workflow rules and all associated workflow actions, such as alerts and field updates. However, any time-based triggers are not included in the package. Notify your installers to set up any time-based triggers that are essential to your app.
- Workflow alerts, field updates, outbound messages, and tasks can be protected by the developer. For more information on protected components, see "Protected Components" in the Salesforce online help.
- Developers can associate or disassociate workflow actions with a workflow rule at any time. These changes, including disassociation, are reflected in the subscriber's organization upon install. In managed packages, a subscriber cannot disassociate workflow actions from a workflow rule if it was associated by the developer.
- References to a specific user in workflow actions, such as the email recipient of a workflow email alert, are replaced by the user installing the package. Workflow actions referencing roles, public groups, account team, sales team, or case team roles may not be uploaded.
- On install, all workflow rules are inactive. The subscriber must deploy the package or individually activate the workflow rules.

## Component Availability After Deployment

Many components have an attribute that controls whether they are available for end users. After installation, certain components are immediately available if they were available in the developer's organization. Other components are not available by default and must be activated using the package deploy process or they can be activated individually.

The following components are available by default based on the developer's setting.

- Apex triggers
- Email templates
- Folders



**Note:** Folders shared with specific users or groups are installed with this option selected, however, no users or groups are selected in the subscriber's organization.

- Letterheads
- Record types
- Validation rules

The following components are not available by default, and must be activated using the package deploy process or they can be activated individually. For more information, see "Deploying Installed Packages" in the Salesforce online help.

- Custom objects
- Custom report types
- Workflow rules

## Protected Components

Developers can mark certain components as *protected*. Protected components cannot be linked to or referenced by components created in a subscriber organization. A developer can delete a protected component in a future release without worrying about

failing installations. However, once a component is marked as unprotected and is released globally, the developer cannot delete it. Developers can mark the following components as protected in managed packages:

- Custom labels
- Custom links (for Home page only)
- Workflow alerts
- Workflow field updates
- Workflow outbound messages
- Workflow tasks

## Intellectual Property Considerations

The following information is important when considering your intellectual property and its protection.

- Only publish package components that are your intellectual property and that you have the rights to share.
- Once components are available on Force.com AppExchange, you cannot recall them from anyone who has installed them.
- The information in the components you package and publish may be visible to users on Force.com AppExchange. Use caution when adding your code to a custom s-control, formula, or any other component that you cannot hide in your app.
- The code contained in an Apex script that is part of a managed package is automatically obfuscated and cannot be viewed in an installing organization. The only exceptions are methods declared as global, meaning that the method signatures can be viewed in an installing organization.

## Publishing Your Apps on Force.com AppExchange

Available for: **Enterprise**, **Unlimited**, and **Developer** Editions

User Permissions Needed	
To install or uninstall packages from AppExchange	"Download AppExchange Packages"
To create new AppExchange packages	"Create AppExchange Packages"
To create test drives and publish AppExchange packages to AppExchange	"Upload AppExchange Packages"

When you create an app for Force.com AppExchange, you can decide how widely you want to share it. You can choose to share it publicly or privately and, optionally, password protect it.

Uploading an app makes it privately available. Salesforce sends you an installation link after your upload is complete, which you can manually distribute to other users who may want to install it. Uploaded apps are like unlisted phone numbers. They do not appear in any categories or in search results on AppExchange. If you give friends the link for one of your uploaded apps, they can install the app directly. This is called private sharing. Also, like unlisted phone numbers, if someone has the link, they can access the package.

A second layer of protection is provided in the form of a password. You can protect a package with a password when it is uploaded so that only those with the password have access to the package contents. Salesforce prompts you for a password when installing an uploaded or published app.

To share your app privately, follow these steps:

- Upload your app (optionally, create a password)
- Distribute the link for your app to installers

To share your app publicly, follow these steps:

- Upload your app (optionally, create a password)
- Register your app
- Create a test drive
- Publish your app

Go to [sites.force.com/appexchange/publisherHome](https://sites.force.com/appexchange/publisherHome) for information on registering, creating a demo, and publishing your app.

## Protecting Your Intellectual Property

Users who install your app will be able to examine all of the details within it, except for the following protections on intellectual property in managed packages.

- The code contained in an Apex script that is part of a managed package is automatically obfuscated and cannot be viewed in an installing organization. The only exceptions are methods declared as global, meaning that the method signatures can be viewed in an installing organization.
- Unlike Apex classes, the content of a Visualforce page in a managed package is not hidden when the package is installed. However, custom controllers, controller extensions, and custom components are hidden.

The details of your custom objects, custom links, reports and all other items installed are as completely revealed to installers as they are in any other Salesforce organization. This gives all administrators complete visibility to gain absolute clarity on these components in your app. While this is one way an installer can examine an app and all its components for any malicious content, it prevents developers from protecting some intellectual property.

## Designing Custom Profile Settings

When designing your app for Force.com AppExchange, create custom profiles for the various users of your app. Give each custom profile the appropriate level of access to your app such as tab visibility or field-level security settings for approval fields on custom objects. Use the following tips when creating custom profiles for apps you want to publish:

- Give your custom profiles a meaningful name that makes it obvious that the profile applies to the app. For example, if you are creating a Human Resources app, you may want to name a custom profile "Approving Manager" for managers that approve requisitions.
- If your custom profiles have a hierarchy, make sure the name indicates where in the hierarchy that profile is. For example, a senior-level manager's profile could have the name "Level 2 Approving Manager."
- Avoid giving your custom profiles names that may be interpreted differently in other organizations. For example, "Level 2 Approving Manager" is a more appropriate profile name than "Sr. Manager."
- The profile `Description` is displayed to the user installing your app. Provide a meaningful description for each custom profile so that users installing your app can view it.

When you package an app, you can include your custom profiles. When a customer installs the app, they will be able to overlay these profile settings into their existing profiles. Only the portions of the profile that apply to your package will be affected.



**Note:** The security settings associated with the standard objects and other existing custom objects in an installer's organization are unaffected.



## Tips for Building Your App

If you have already developed your app on the Force.com platform, review these steps as a checklist. If not, use the Salesforce online help for instructions on building a custom app and review the following steps as guidelines.

### Develop app documentation

Provide documentation for your app, including custom help.

### Manage license agreements

Get some tips on handling a license agreement.

### Work with external applications

If your app connects to an external service, get some tips for launching it within Salesforce.

### Provision external services

If your app connects to an external service, get tips on provisioning.

### Uninstall apps

Consider how users will uninstall your app.

### Match the Salesforce user interface

Model the look and feel of your app after the Salesforce user interface.

## Developing App Documentation

Available in: <b>Group</b> , <b>Professional</b> , <b>Enterprise</b> , <b>Unlimited</b> , and <b>Developer</b> Editions
---

User Permissions Needed	
To edit custom tabs:	"Customize Application"

Salesforce recommends publishing your app on Force.com AppExchange with the following types of documentation:

### About Tab

Most apps are installed by an administrator. Therefore, most users are not aware of an app until it is deployed. After deployment, an About tab is very useful to users and can give them an introduction to the entire app.

An About tab is a custom tab included with an app on AppExchange. It provides users with:

- A description of the app
- Training and support information

Providing users with training options is the best way to ensure customer adoption of your app. Similar to the way that salesforce.com provides free training, you should provide user training for commercial applications either through webinars or recorded presentations. Include a section in your About tab that covers how to get training on the app.

If users of your app call salesforce.com for assistance, we will try to determine if the issue resides with our application or yours. We can only support our application, our standard user interface, and the installation of apps from AppExchange. If the customer is asking about functionality that is unique to your app, we will ask them to examine the About tab for information on how to get support for your app. We expect commercial apps to provide support that is similar to the support salesforce.com offers and that the About tab contain contact information for this support.



**Note:** An About tab adds to an organization's total number of custom tabs. Consider Salesforce Editions and Limits for each Edition before creating an About tab that could make an installer exceed the limit of allowed custom tabs.

### Configure Option

You can include a **Configure** option for installers of your app when they view AppExchange installs and package detail pages in their Salesforce organization. This option can link to installation and configuration details such as:

- Adding custom fields and custom links to standard objects
- Requesting or executing an end user license agreement
- Provisioning the external service of a composite app

The **Configure** option is included in your package as a custom link. You can create a custom link for your home page layouts and add it to your package. To do this:

1. Create a custom link to a URL that contains configuration information or an s-control or Visualforce page that implements configuration. When you create your custom link, set the display properties to `Open in separate popup window` so that the user returns to the same Salesforce page when done.
2. When you create the package, choose this custom link in the `Configure Custom Link` field of your package detail.

### Multimedia Presentation

If you are publishing a commercial app, we recommend compiling a presentation, such as a Breeze presentation, to demonstrate its attributes.

### Data Sheet

Give installers the fundamental information they need to know about your app before they install.

### Customization and Enhancement Guide

Let installers know what they should customize after installation as part of their implementation.

### Custom Tab Splash Page

You can add a splash page to any custom tabs in your app. Custom splash pages can include information about using the custom tab and custom object with links to additional information. They include a **Continue** button and a **Don't show me this again** button.

Create the splash page for your app in the form of a custom link for the Home tab via **Setup** ► **Customize** ► **Home** ► **Custom Links**. Custom links can be a URL to an HTML page or a custom s-control or Visualforce page. Create your custom splash page as a Home tab custom link and then add it as the splash page option to any custom tabs you plan to upload to AppExchange.

### Custom Help

You can provide custom help for your custom object records and custom fields.



**Tip:** To give your custom help a professional tone using Salesforce terminology, follow the [Salesforce.com Style Guide for Documentation and User Interface Text](#). To use the Salesforce styles, reference the Salesforce online help style sheet with a relative path of `/sCSS/Theme2/default/help.css`.

## Managing License Agreements

Force.com AppExchange provides a standard mechanism for presenting a license agreement to the installer and requiring acceptance of this agreement before continuing with the installation. If you want to require an agreement, paste the text of your license agreement in the `License Agreement` text field on the app listing page on AppExchange directory.

If your app requires a more interactive license negotiation, make sure your About tab splash page clearly details the required end user license agreement and states your preferred process for getting started. In all cases, salesforce.com includes a disclaimer that limits our liability for the app.

## Working with External Services

You may want to update your Salesforce data when changes occur in another service. Likewise, you may also want to update the data in another service (outside of Salesforce) based on changes to your Salesforce data. Salesforce provides ways of doing both of these transactions. For example, you may want to send a mass email to more contacts and leads than Salesforce allows. To do so, you can use an external mail service that allows users to build a recipient list of names and email addresses using the contact and lead information in your Salesforce organization.

An app built on the Force.com platform can connect with an external service in many different ways. For example:

- You can create a custom link or custom formula field that passes information to an external service.
- You can use the Force.com API to transfer data in and out of Salesforce.
- You can use an Apex class that contains a Web service method.



**Caution:** Do not store usernames and passwords within any external service.

## Provisioning External Services

If your app links to an external service, users who install the app must be signed up to use the external service. Provide access to your external service in one of two ways:

- Access by all active users in an organization with no real need to identify an individual
- Access on a per user basis where identification of the individual is important

The Salesforce service provides two globally unique IDs to support these options. The user ID identifies an individual and is unique across all organizations. User IDs are never reused. Likewise, the organization ID uniquely identifies the organization.

Avoid using email addresses, company names, and Salesforce usernames when providing access to an external service. Usernames can change over time and email addresses and company names can be duplicated.

If you are providing access to an external service, we recommend the following:

- Use Single Sign-On (SSO) techniques to identify new users when they use your service.

- For each point of entry to your app, such as a custom link or web tab, include the user ID in the parameter string. Have your service examine the user ID to verify that the user ID belongs to a known user. Include a session ID in the parameter string so that your service can read back through the Force.com API and validate that this user has an active session and is authenticated.
- Offer the external service for any known users. For new users, display an alternative page to collect the required information for the user.
- Do not store passwords for individual users. Besides the obvious security risks, many organizations reset passwords on a regular basis, which requires the user to update the password on your system as well. We recommend designing your external service to use the user ID and session ID to authenticate and identify users.
- If your application requires asynchronous updates after a user session has expired, dedicate a distinct administrator user license for this.

## Uninstalling Apps

Users can remove an installed app following the uninstall instructions in the Salesforce online help. By doing an uninstall, users can remove all the components of an app in their organization with a single click and confirmation. However, the publisher of the app will not be notified when an uninstall occurs unless the app is based on a managed package and the publisher has installed the License Management App.

## Matching the Salesforce User Interface

Apps that resemble the Salesforce user interface look and feel are instantly more familiar to users and easy to use. The easiest way to model the design of your app after the Salesforce user interface look and feel is to use Visualforce. When you use a standard controller with a Visualforce page, your new page takes on the style of the associated object's standard tab in Salesforce. For more information, see [Using Salesforce Styles](#).

You can also use the style sheets and design guidelines from salesforce.com to present a familiar interface to users of your app. The Salesforce style sheets and design guidelines are available as downloads from [sites.force.com/appexchange](https://sites.force.com/appexchange). For more information on the Salesforce user interface styles, see [Using the Salesforce CSS in Your Apps](#).

## Packaging and Uploading Apps

After you have finished building your app and preparing it for Force.com AppExchange, it is ready to share. Follow the steps in the next section to package all the components in your app and upload it.

If you want to post an app listing on AppExchange and share it publicly, go to [sites.force.com/appexchange/publisherHome](https://sites.force.com/appexchange/publisherHome) for information on registering and posting it.

## Preparing Your Apps for Distribution

Available in: **Group**, **Professional**, **Enterprise**, **Unlimited**, and **Developer** Editions

User Permissions Needed	
To create packages:	"Create AppExchange Packages"
To upload packages:	"Upload AppExchange Packages"

An app can contain many different components and you can create, upload, and register your apps on your own timeline. To group components in a container to upload to Force.com AppExchange, create a package and add components to the package. A package is the container for the app that you must use to upload all its components together. Before you begin, determine if you want to upload a managed or unmanaged package. For more information about managed packages, see "About Managed Packages" in the Salesforce online help.

1. Create a package:
  - a. Click **Setup** ► **Create** ► **Packages**.
  - b. Click **New**.
  - c. Enter a name for your package. This does not have to be the same name that appears on AppExchange.
  - d. Optionally, choose a custom link from the `Configure Custom Link` field to display configuration information to installers of your app. You can select a predefined custom link to a URL or s-control that you have created for your home page layouts; see the "Configure Option" in the Salesforce online help. The custom link displays as a **Configure** link within Salesforce on the Force.com AppExchange Downloads page and app detail page of the installer's organization.
  - e. Optionally in the `Apex Exception Notification Recipient` field, enter the username of the person who should receive an email notification if an exception occurs in an Apex script that is not caught by the script. If you do not specify a username, all uncaught exceptions generate an email notification that is sent to salesforce.com. This is only available for managed packages. For more information, see "Handling Apex Exceptions in Managed Packages" in the Salesforce online help.



**Note:** Apex can only be packaged from Developer, Enterprise, and Unlimited Edition organizations.

- f. Optionally, enter a description that describes the package. You will have a chance to change this description before you upload it to AppExchange.
- g. Click **Save**.

2. Salesforce sets your package API access privileges to `Unrestricted`. You can change this setting to further restrict API access of Salesforce components in the package. For more information, see "Managing API and Dynamic Apex Access in Packages" in the Salesforce online help.
3. Add the necessary components for your app.


- a. Click **Add**.

- b. From the drop-down list, choose the type of component you want to add to your package.

- At the top of the list, click a letter to display the contents of the sorted column that begin with that character.
- If available, click the **Next Page** (or **Previous Page**) link to go to the next or previous set of components.
- If available, click **fewer** or **more** at the bottom of the list to view a shorter or longer display list.

- c. Select the components you want to add.



**Note:** Some components cannot be added to  Managed - Released packages. For a list of these components, see "Packageable Components" in the Salesforce online help.

S-controls cannot be added to packages with restricted API access.

- d. Click **Add To Package**.

Repeat these steps until you have added all the components you want in your package.



**Note:** Some related components are automatically included in the package even though they may not display in the Package Components list. For example, when you add a custom object to a package, its custom fields, page layouts, and relationships with standard objects are automatically included. For a complete list of components Salesforce automatically includes, see "Implicitly Included Packaged Components" in the Salesforce online help.

4. Optionally, click **Show dependencies** and review a list of components that rely on other components, permissions, or preferences within the package. An entity may include such things as an s-control, a standard or custom field, or an organization-wide setting like multicurrency. Your package cannot be installed unless the installer has the listed components enabled or installed. See "Understanding Dependencies" in the Salesforce online help for more information on dependencies. Click **Done** to return to the Package detail page.



**Note:** You cannot upload packages that contain any of the following:

- References to person accounts, such as an s-control or custom field referencing person accounts.
- Workflow rules or workflow actions (such as field updates or outbound messages) that reference record types.
- Reports that reference record types on standard objects.

5. Click **Upload**.



**Note:** If you are creating a managed package to publish on AppExchange, you must certify your application before you package it. To start the certification process, see [Security Review](#).

On the Upload Package page, do the following:

- a. Enter a `Version Name`, such as `June 2009`. This field is required for managed and unmanaged packages. The version name is the marketing name for a specific release of a package and allows you to create a more descriptive title for the package.
- b. Enter a `Version Number` for the upload, such as `1.0`. The format is `majorNumber.minorNumber`.

The version number represents a release of a package. This field is required for managed and unmanaged packages. For a managed package, the version number corresponds to a Managed - Released upload. All Managed - Beta uploads use the same version number until you upload a Managed - Released package version with a new version number. For example, the following is a sequence of version numbers for a series of uploads.

Upload Sequence	Type	Version Number	Notes
First upload	Managed - Beta	1.0	The first Managed - Beta upload.
Second upload	Managed - Released	1.0	A Managed - Released upload. Note that the version number does not change.
Third upload	Managed - Released	1.1	Note the change of minor release number for this Managed - Released upload.
Fourth upload	Managed - Beta	2.0	The first Managed - Beta upload for version number 2.0. Note the major version number update.
Fifth upload	Managed - Released	2.0	A Managed - Released upload. Note that the version number does not change.

c. For managed packages, select a `Release Type`:

- Choose **Managed - Released** to upload an upgradeable version. After upload, some attributes of Salesforce components are locked. For a list of locked attributes, see "Component Attributes in Managed Packages" in the Salesforce online help.
- Choose **Managed - Beta** if you want to upload a version of your package to a small sampling of your audience for testing purposes. You will still be able to change the components and upload additional beta versions. For information on beta versions, see "Beta Versions of Managed Packages" in the Salesforce online help.



**Note:** Beta packages can only be installed in Developer Edition or sandbox organizations.

d. Change the `Description`, if necessary.

e. Optionally, enter and confirm a password to share the package privately with anyone who has the password. Do not enter a password to make the package available to anyone on AppExchange and share your package publicly.

f. Salesforce automatically selects the requirements it finds. In addition, select any other required components from the `Package Requirements` and `Object Requirements` sections to notify installers of any requirements for this package.

g. Click **Continue**.

6. Once your upload is complete you can do any of the following.

- Click **Edit Security** to change the password option.



**Tip:** If you have forgotten the password for an upload, contact salesforce.com and provide the ID listed on the `Edit Security` page.

- Click **Deprecate** to prevent new installations of this package while allowing existing installations to continue operating.



**Note:** You cannot deprecate the most recent version of an uploaded managed package.

When you deprecate a package, remember to remove it from AppExchange as well. See *"Removing Apps from AppExchange"* in the Force.com AppExchange online help.

- Click **Make Active** to make a deprecated upload available for installation again.



**Note:**

After you upload a new version of your Managed - Released package, you can click **Deprecate** so installers cannot install an older version. Deprecation prevents new installations of older versions without affecting existing installations. For more information, see "Managing Uploads" in the Salesforce online help.

You cannot deprecate the most recent version of a managed package upload.

You will receive an email that includes an installation link when your package has been uploaded successfully. Wait a few moments before clicking the installation link or distributing it to others, as it may take a few minutes for it to become active.



**Note:** If you uploaded from your Salesforce production organization, notify installers who want to install it in a sandbox organization to replace the "www.salesforce.com" portion of the installation link with "test.salesforce.com". Likewise, if you uploaded from a sandbox organization, notify installers who want to install it in their Salesforce production organization to replace the "test.salesforce.com" portion of the installation link with "www.salesforce.com". If you published an package from your sandbox, you will only be able to manage it until you create a new copy. Once you create a new copy of your sandbox, you will no longer be able to manage packages that you uploaded from it.



## **Publishing and Listing Your App**

After your app is registered, you can publish it along with a listing on Force.com AppExchange. Go to [sites.force.com/appexchange/publisherHome](https://sites.force.com/appexchange/publisherHome) for information on publishing your app on Force.com AppExchange.

## Supporting Apps

After your app is available on Force.com AppExchange, provide support for it using Salesforce and AppExchange:

### Provide end user support

Be prepared for users who may call requesting support.

### Promote your app

Get your app the attention it needs.

### Find out how to upgrade

Provide a solution for upgrades to your app.

### Monitor usage of your app

Track who has installed your app.

### Bill for use of your app

Currently, salesforce.com does not support billing users for partner apps.

### View a checklist for success

Consider the dos and don'ts of posting your app.

### Frequently asked questions

Consult the frequently asked questions on the AppExchange website.

## Providing End User Support

The publisher is responsible for providing end user support of all published apps. When customers contact salesforce.com Customer Support with a question about your app's business processes or the user interface, we will direct the user to find support information on the About tab of your app. Make sure the About tab for your app includes this information.

## Promoting Your App


Through our Force.com AppExchange partner programs, we offer opportunities for developers of AppExchange apps to jointly promote their apps with salesforce.com.

## Upgrading Packages

Available in: **Group, Professional, Enterprise, Unlimited, and Developer** Editions

User Permissions Needed	
To upload packages:	"Upload AppExchange Packages"
To uninstall packages:	"Download AppExchange Packages"
To install packages:	"Download AppExchange Packages"

Salesforce supports upgrades for managed packages only. Publishers can publish an upgrade for a managed package and notify installers that the new version is available. Installers of a managed package can then install the upgrade as follows:

1. Before you install an upgrade, determine if the app you installed was from a managed package. Look for the  Managed - Installed icon on the detail pages for each component and on the list of packages installed.  
If the app you installed is not from a managed package, upgrades for it are not available.
2. Then, install the upgrade in the same way you would install any other package from Force.com AppExchange directory. If the publisher provided a link to the new version, follow the link to the package posting and install it in your organization. The first page of the install wizard lists the current version you have installed, the version you are about to install, and a list of additional components included in the new version.

### Notes on Upgrading Managed Packages

Consider the following when upgrading a managed package:

- All existing custom objects that were previously deployed will still be deployed. Salesforce prompts you to deploy any new custom objects or previously undeployed custom objects.
- Profile settings are editable but not upgradeable. After upgrading, update the profile settings for any custom objects included in the package. The developer can make changes to profile settings after releasing them, but those changes will not be included in an upgrade.
- If the developer chooses to add universally required custom fields, the fields will have default values.
- Translation workbench values for components that are "editable but not upgradeable" based on the Packageable Components are excluded from upgrades.
- If an installed package has `Restricted API` access, upgrades will be successful only if the upgraded version does not contain any s-controls. If s-controls are present in the upgraded version, you must change the currently installed package to `Unrestricted API` access.
- When you upgrade a package, changes to the API access are ignored even if the developer specified them. This ensures that the administrator installing the upgrade has full control. Installers should carefully examine the changes in package access in each upgrade during installation and note all acceptable changes. Then, because those changes are ignored, the administrator should manually apply any acceptable changes after installing an upgrade.

## Monitoring Usage of Your App

Currently, salesforce.com does not support a formal mechanism for informing AppExchange publishers about the number of times their apps have been installed. However, if your app is based on a managed package, you must have the License Management App installed. The License Management App tracks the number of licenses that are currently installed and uninstalled for your app. It does not track information for unmanaged packages.

## Billing for Use of Your App

Currently salesforce.com does not support billing on behalf of partners. We may offer this service at a future date.

## Checklist for Success

Use the following list when preparing, building, and posting your app on AppExchange:

### Do

- Try to match the Salesforce user interface. For information on detecting the user interface theme and adhering to Salesforce user interface styles, see [Using the Salesforce CSS in Your Apps](#).
- Provide online help when possible or screens that document the app's intended use clearly.
- Upload your documentation files to the Documents tab and store them in a folder that is specific to your app. Include help and support keywords for these documents so that they show in search results.
- Review all test drive data, including text and images, for content that might be considered offensive.

### Do Not

- Require a browser setting that is not standard.
- Restrict your app to a single browser as this can increase support costs and efforts.
- Store usernames and passwords in your external service. Instead, use `User ID` to identify the actual user and `Session ID` to retrieve the necessary information.
- Use real contact information as test drive data.
- Use the word "force" in your app name or label.
- Target your presentation to salesforce.com. Your target audience should be your intended customer.
- Publish apps with deprecated API calls. Always use the latest available API.
- Create a test drive that attempts to store new data or change data to existing records. Because users viewing your test drive have read-only capability, changes made outside of Salesforce but within the test drive could fail.

## Frequently Asked Questions

See [sites.force.com/appexchange](https://sites.force.com/appexchange) for answers to the most frequently asked questions.

# Index

## A

- App
  - publish 20
- AppExchange
  - apps 9
  - creating packages 27
  - custom help 23
  - designing apps 13
  - developing app documentation 23
  - uploading packages 27

## Apps

- design 13
- uploading 27

## C

- Custom help
  - AppExchange apps 23

- Custom help (*continued*)
  - style guide 25

## F

- Force.com AppExchange
  - overview 5

## P

- Packages
  - creating 27
  - upgrading packages 33
  - uploading 27
- Publish
  - apps 20