App management

In this section, you can set policies related to app availability, installation, updates, and permission management.

> Managed Google Play Accounts are automatically created when devices are provisioned.

1. Play Store mode

This mode controls which apps are available to the user in the Play Store and the behavior on the device when apps are removed from the policy.

> **Whitelist (default)**: Only apps that are in the policy are available and any app not in the policy will be automatically uninstalled from the device. The Play Store will only show available apps.
>
> **Blacklist**: All apps are available and any app that should not be on the device should be explicitly marked as **blocked** in the applications policy. The Play Store will show all apps, except blocked ones.

2. Untrusted apps policy

The policy for untrusted apps (apps from unknown sources) enforced on the device. This option controls the Android system setting that determines whether a user can install apps from outside the Play Store (sideloading).

> **Disallow (default)**: Disallow untrusted app installs on the entire device.
>
> **Personal profile only**: For devices with work profiles, allow untrusted app installs in the device's personal profile only.
>
> **Allow**: Allow untrusted app installs on the entire device.

3. Google Play Protect

Whether Google Play Protect app verification is enforced.

Enforced (default): Force-enables app verification.

User choice: Allows the user to choose whether to enable app verification.

4. Default permission policy

The policy for granting runtime permission requests to apps.

Prompt (default): Prompt the user to grant a permission.

Grant: Automatically grant a permission.

Deny: Automatically deny a permission.

5. App functions

Controls whether apps on fully managed devices or in work profiles are allowed to expose app functions. Requires Android 16 or above.

Allowed (default): Apps on fully managed devices or in work profiles can expose app functions.

Disallowed: Apps on fully managed devices or in work profiles cannot expose app functions.

6. Install apps disabled

Whether user installation of apps is disabled.

7. Uninstall apps disabled

Whether user uninstallation of applications is disabled.

8. Permission policies

Explicit permission or group grants or denials for all apps. These values override the **Default permission policy** setting.

Use **Add permission policy** to create entries and remove them with the delete action.

Each entry includes:

> **Android permission/group**: The Android permission or group (required), for example **android.permission.READ_CALENDAR** or **android.permission_group.CALENDAR**.
>
> **Policy**: Grant / Deny / Prompt (uses the same policy options as **Default permission policy**).

9. Applications

List of applications that must be included in the policy. The behavior of the list's content depends on the value set on **Play Store mode**.

If **Play Store mode** is set to **whitelist**, only apps that are in the policy are available and any app not in the policy will be automatically uninstalled from the device.

If **Play Store mode** is set to **blacklist**, all apps are available and any app that should not be on the device should be explicitly marked as **blocked** in the applications policy.

To add a new app, click on the **Add applications** button (or the **Add applications** icon), select **From managed Google Play**, then choose the app from Play Store and click on the **Select** button in the app card.

To add an app that is not published on Google Play but is already present on the device (for example, a system or manufacturer app), click on **Add applications**, select **By package name (preinstalled apps)**, type the app's package name (e.g. **com.example.app**) and click on **Add** .The app is added with the **Force installed** install type. To show the app in the kiosk custom launcher, its **Install type** must be **Preinstalled** or **Force installed**: with **Available** the app isn't shown.

> Apps that are not available on managed Google Play are marked with an Android icon in the applications list. They can't be installed from Google Play, so they must already be present on the device: devices that don't have the app can't install it.

> All apps that are published on the Play Store in your country are available for selection by default. To select your own private or web apps, you must upload them to the system first.
>
> For more information read the **Private apps** page.

Each app can be configured with its own settings, that are visually contained in a card:

9.1. Install type

The type of installation to perform for an app.

Available: The app is available to install.

Preinstalled: The app is automatically installed and can be removed by the user.

Force installed: The app is automatically installed and can't be removed by the user.

Blocked: The app is blocked and can't be installed. If the app was installed under a previous policy, it will be uninstalled.

Required for setup: The app is automatically installed and can't be removed by the user and will prevent setup from completion until installation is complete.

Kiosk: The app is automatically installed in kiosk mode: it's set as the preferred home intent and whitelisted for lock task mode. Device setup won't complete until the app is installed. After installation, users won't be able to remove the app. You can only set this **install type** for one app per policy. When this is present in the policy, status bar will be automatically disabled. For more information please read the dedicated **Kiosk mode** page.

9.2. Install constraints

Defines a set of restrictions for the app installation. When multiple constraints are selected, all of them must be satisfied for the app to be installed.

> This option is shown only when the **Install type** is **Preinstalled** or **Force installed**.

Unmetered network: Install the app only when the device is connected to an unmetered network (e.g. Wi-Fi).

Charging: Install the app only when the device is charging.

Idle: Install the app only when the device is idle.

9.3. Auto-update mode

Controls the auto-update mode for the app.

Default: The app is automatically updated with low priority to minimize the impact on the user. The app is updated when all of the following constraints are met: (1) the device is not actively used, (2) the device is connected to an unmetered network, (3) the device is charging. The device is notified about a new update within 24 hours after it is published by the developer, after which the app is updated the next time the constraints above are met.

Postponed: The app is not automatically updated for a maximum of 90 days after the app becomes out of date. 90 days after the app becomes out of date, the latest available version

is installed automatically with low priority (see **Default** Auto-update mode). After the app is updated, it is not automatically updated again until 90 days after it becomes out of date again. The user can still manually update the app from the Play Store at any time.

High priority: The app is updated as soon as possible. No constraints are applied. The device is notified immediately about a new update after it becomes available.

9.4. Minimum version code

The minimum version of the app that runs on the device. If set, the device attempts to update the app to at least this version code. If the app is not up-to-date, the device will contain a **Non-compliance detail** with **Non-compliance reason** set to **APP_NOT_UPDATED**. The app must already be published to Google Play with a version code greater than or equal to this value. At most 20 apps may specify a minimum version code per policy.

9.5. Delegated scopes

The scopes delegated to the app from Android Device Policy. You can grant other apps a selection of special Android permissions:

Certificate installation: Grants access to certificate installation and management.

Managed configurations: Grants access to managed configurations management.

Block uninstall: Grants access to blocking uninstallation.

Permissions: Grants access to permission policy and permission grant state.

Package access: Grants access to package access state.

System app: Grants access for enabling system apps.

9.6. Preferential Network

The preferential network service to use for this app. If set, the app will use the specified enterprise network slice for its connections when available. This must match a network slice configured in the **5G Network Slicing Configuration** section of the **Cellular** panel.

9.7. Default permission policy

The default policy for all permissions requested by the app. If specified, this overrides the policy-level **Default permission policy** which applies to all apps. It does not override the **Permission policies** which applies to all apps.

Prompt (default): Prompt the user to grant a permission.

Grant: Automatically grant a permission.

Deny: Automatically deny a permission.

9.8. Connected work and personal app

Controls whether the app can communicate with itself across a device's work and personal profiles, subject to user consent (Android 11+).

Disallowed (default): Prevents the app from communicating cross-profile.

Allowed: Allows the app to communicate across profiles after receiving user consent.

9.9. Always On VPN lockdown exemption

Specifies whether the app is allowed networking when the VPN is not connected and **lockdown enabled** is active. Only supported on devices running Android 10 and above.

Enforced (default): The app respects the always-on VPN lockdown setting.

Exempt: The app is exempt from the always-on VPN lockdown setting.

9.10. Work profile widgets

Specifies whether the app installed in the work profile is allowed to add widgets to the home screen.

Allowed: The application can add widgets to the home screen.

Disallowed: The application cannot add widgets to the home screen.

9.11. User control settings

Specifies whether user control is permitted for a given app. User control includes user actions like force-stopping and clearing app data (Android 11+). If **extensionConfig** is enabled for an app, user control is disallowed regardless of this setting. For kiosk apps, you can use **Allowed** to allow user control.

Unspecified: Uses the default behavior of the app to determine if user control is allowed or disallowed.

Allowed: User control is allowed for the app.

Disallowed: User control is disallowed for the app.

9.12. Disabled

Whether the app is disabled. When disabled, the app data is still preserved.

9.13. Allow Credential Provider

Whether the app is allowed to act as a credential provider on Android 14 and above.

9.14. Managed configuration

To configure the app's managed settings, click on the **Enable managed configuration** button. If a managed configuration is already set for the app, you can modify the configuration with the **Managed configuration** button, or delete it with the **Remove configuration** button.

> **Managed configuration** option is available only for apps that supports this functionality.

9.15. Permission policies

Explicit permission grants or denials for the app. These values override the **Default permission policy** and **Permission policies** which apply to all apps.

Use **Add permission policy** to add one or more permission rules for the app card and remove them with the delete action.

9.16. Track IDs

List of the app’s closed testing track IDs that a device can access. If multiple track IDs are selected, devices receive the latest version among all accessible tracks. If no track IDs is selected, devices only have access to the app’s production track.

> **Track IDs** option is available only for apps that have at least one track ID available for your organization. For more details on how to add your organization to a closed testing track for a specific app please read here.

10. Default application settings

Set default apps for supported types. When a default app is set for at least one type, users are prevented from changing default apps in that profile.

> Only one default application setting is allowed per **Default application type**. The list of default applications must not contain duplicates.

10.1. Default application type

Select the app category to configure (for example Browser, Dialer, SMS, Wallet, or Assistant). Availability depends on Android version and management mode.

10.2. Default application scopes

Select where the default app should apply (Fully managed, Work profile, or Personal profile). Only scopes supported by the selected type can be chosen.

> If none of the selected scopes are applicable to the device’s management mode, the device reports a non-compliance detail.

10.3. Default applications

List of apps that can be set as default for the selected type. The first installed and eligible app is set as the default.

> If scopes include **Fully managed** or **Work profile**, each app must also exist in the **Applications** list with **Install type** not set to **Blocked**.

11. Private key selection

Allows showing UI on a device for a user to choose a private key alias if there are no matching rules in **Choose private key rules**.

> For devices below Android P, setting this may leave enterprise keys vulnerable.

12. Choose private key rules

Controls apps' access to private keys. The rule determines which private key, if any, Android Device Policy grants to the specified app. Access is granted either when the app calls KeyChain.choosePrivateKeyAlias (or any overloads) to request a private key alias for a given URL, or for rules that are not URL-specific (that is, if urlPattern is not set, or set to the empty string or .*) on Android 11 and above, directly so that the app can call KeyChain.getPrivateKey, without first having to call KeyChain.choosePrivateKeyAlias. When an app calls KeyChain.choosePrivateKeyAlias if more than one choosePrivateKeyRules matches, the last matching rule defines which key alias to return.

Use **Add private key rule** to create entries and remove them with the delete action.

12.1. Private key alias

The alias of the private key to be used.

12.2. URL pattern

The URL pattern to match against the URL of the request. If not set or empty, it matches all URLs. This uses the regular expression syntax of java.util.regex.Pattern.

12.3. Package names

The package names to which this rule applies. The hash of the signing certificate for each app is verified against the hash provided by Play. If no package names are specified, then the alias is provided to all apps that call KeyChain.choosePrivateKeyAlias or any overloads (but not without calling KeyChain.choosePrivateKeyAlias, even on Android 11 and above). Any app with the same Android UID as a package specified here will have access when they call KeyChain.choosePrivateKeyAlias.

Use **Add package name** to add entries and remove them with the delete action.

> To delete an app, click on the **trashbin** icon, on the bottom of the app's card.

Revision #79
Created 2022-03-22 09:44:05 UTC by Admin
Updated 2026-11-12 09:52:10 UTC by Admin