Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

DeviceAdminReceiver

Kotlin (/reference/kotlin/android/app/admin/DeviceAdminReceiver) | **Java**

`public class DeviceAdminReceiver`
`extends BroadcastReceiver` (/reference/android/content/BroadcastReceiver)

java.lang.Object (/reference/java/lang/Object)
↳ android.content.BroadcastReceiver (/reference/android/content/BroadcastReceiver)
↳ android.app.admin.DeviceAdminReceiver

Base class for implementing a device administration component. This class provides a convenience for interpreting the raw intent actions that are sent by the system.

The callback methods, like the base `BroadcastReceiver.onReceive()` (/reference/android/content/BroadcastReceiver#onReceive(android.content.Context,%20android.content.Intent)) method, happen on the main thread of the process. Thus long running operations must be done on another thread. Note that because a receiver is done once returning from its receive function, such long-running operations should probably be done in a `Service` (/reference/android/app/Service).

When publishing your DeviceAdmin subclass as a receiver, it must handle `ACTION_DEVICE_ADMIN_ENABLED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ADMIN_ENABLED) and require the `Manifest.permission.BIND_DEVICE_ADMIN` (/reference/android/Manifest.permission#BIND_DEVICE_ADMIN) permission. A typical manifest entry would look like:

```
<receiver android:name=".app.DeviceAdminSample$DeviceAdminSampleReceiver"
        android:label="@string/sample_device_admin"
        android:description="@string/sample_device_admin_description"
        android:permission="android.permission.BIND_DEVICE_ADMIN">
```

```
    <meta-data android:name="android.app.device_admin"
               android:resource="@xml/device_admin_sample" />
    <intent-filter>
        <action android:name="android.app.action.DEVICE_ADMIN_ENABLED" />
    </intent-filter>
</receiver>
<receiver android:name=".app.DeviceAdminSample$DeviceAdminSampleReceiver2"
    android:label="@string/sample_device_admin2"
    android:description="@string/sample_device_admin_description2"
    android:permission="android.permission.BIND_DEVICE_ADMIN">
    <meta-data android:name="android.app.device_admin"
        android:resource="@xml/device_admin_sample" />
    <intent-filter>
        <action android:name="android.app.action.DEVICE_ADMIN_ENABLED" />
    </intent-filter>
</receiver>
```

The meta-data referenced here provides addition information specific to the device administrator, as parsed by the `DeviceAdminInfo` (/reference/android/app/admin/DeviceAdminInfo) class. A typical file would be:

```
<device-admin xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-policies>
        <limit-password />
        <watch-login />
        <reset-password />
        <force-lock />
        <wipe-data />
        <expire-password />
        <encrypted-storage />
        <disable-camera />
        <disable-keyguard-features />
    </uses-policies>
</device-admin>
```

Developer Guides

For more information about device administration, read the Device Administration (/guide/topics/admin/device-admin) developer guide.

Summary

Constants

String (/reference/java/lang/String)	ACTION_CHOOSE_PRIVATE_KEY_ALIAS (/reference/android/app/admin/DeviceAdminReceiver#ACTION_CHOOSE Broadcast action: notify that some app is attempting to choose a KeyCha
String (/reference/java/lang/String)	ACTION_DEVICE_ADMIN_DISABLED (/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ Action sent to a device administrator when the user has disabled it.
String (/reference/java/lang/String)	ACTION_DEVICE_ADMIN_DISABLE_REQUESTED (/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ Action sent to a device administrator when the user has requested to dis done.
String (/reference/java/lang/String)	ACTION_DEVICE_ADMIN_ENABLED (/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ This is the primary action that a device administrator must implement to
String (/reference/java/lang/String)	ACTION_LOCK_TASK_ENTERING (/reference/android/app/admin/DeviceAdminReceiver#ACTION_LOCK_TA Action sent to a device administrator to notify that the device is entering
String (/reference/java/lang/String)	ACTION_LOCK_TASK_EXITING (/reference/android/app/admin/DeviceAdminReceiver#ACTION_LOCK_TA Action sent to a device administrator to notify that the device is exiting lo
String (/reference/java/lang/String)	ACTION_NETWORK_LOGS_AVAILABLE (/reference/android/app/admin/DeviceAdminReceiver#ACTION_NETWOR Broadcast action: notify that a new batch of network logs is ready to be

String (/reference/java/lang/String)	ACTION_PASSWORD_CHANGED (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWC Action sent to a device administrator when the user has changed the pas challenge.
String (/reference/java/lang/String)	ACTION_PASSWORD_EXPIRING (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWC Action periodically sent to a device administrator when the device or pro
String (/reference/java/lang/String)	ACTION_PASSWORD_FAILED (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWC Action sent to a device administrator when the user has entered an incor password.
String (/reference/java/lang/String)	ACTION_PASSWORD_SUCCEEDED (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWC Action sent to a device administrator when the user has successfully ent password, after failing one or more times.
String (/reference/java/lang/String)	ACTION_PROFILE_PROVISIONING_COMPLETE (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PROFILE Broadcast Action: This broadcast is sent to indicate that provisioning of a completed successfully.
String (/reference/java/lang/String)	ACTION_SECURITY_LOGS_AVAILABLE (/reference/android/app/admin/DeviceAdminReceiver#ACTION_SECURIT Broadcast action: notify that a new batch of security logs is ready to be c
int	BUGREPORT_FAILURE_FAILED_COMPLETING (/reference/android/app/admin/DeviceAdminReceiver#BUGREPORT_FAIL Bugreport completion process failed.
int	BUGREPORT_FAILURE_FILE_NO_LONGER_AVAILABLE (/reference/android/app/admin/DeviceAdminReceiver#BUGREPORT_FAIL

	Bugreport has been created, but is no longer available for collection.
String (/reference/java/lang/String)	DEVICE_ADMIN_META_DATA (/reference/android/app/admin/DeviceAdm Name under which a DevicePolicy component publishes information abo
String (/reference/java/lang/String)	EXTRA_DISABLE_WARNING (/reference/android/app/admin/DeviceAdmin A CharSequence that can be shown to the user informing them of the im
String (/reference/java/lang/String)	EXTRA_LOCK_TASK_PACKAGE (/reference/android/app/admin/DeviceAdminReceiver#EXTRA_LOCK_TAS A string containing the name of the package entering lock task mode.
String (/reference/java/lang/String)	EXTRA_TRANSFER_OWNERSHIP_ADMIN_EXTRAS_BUNDLE (/reference/android/app/admin/DeviceAdminReceiver#EXTRA_TRANSFEI A Parcelable (/reference/android/os/Parcelable) extra of type Persis (/reference/android/os/PersistableBundle) that allows a mobile device m the management application instance after owner transfer.

Public constructors

DeviceAdminReceiver (/reference/android/app/admin/DeviceAdminReceiver#DeviceAdminReceiver())
( )

Public methods

DevicePolicyManager (/reference/android/app/admin/DevicePolicyManager)	getManager (/reference/android/app/admin/Device Retrieve the DevicePolicyManager interface for this
ComponentName (/reference/android/content/ComponentName)	getWho (/reference/android/app/admin/DeviceAdmin

	Retrieve the ComponentName describing who this d the administrator to identify itself.
`void`	`onBugreportFailed` (/reference/android/app/admi (/reference/android/content/Context) `context, I` Called when the bugreport collection flow has failed
`void`	`onBugreportShared` (/reference/android/app/admi (/reference/android/content/Context) `context, I` Called when the bugreport has been shared with the
`void`	`onBugreportSharingDeclined` (/reference/andro `(Context` (/reference/android/content/Context) `co` Called when sharing a bugreport has been cancelled
`String` (/reference/java/lang/String)	`onChoosePrivateKeyAlias` (/reference/android/app/admin/DeviceAdminReceive `(Context` (/reference/android/content/Context) `co` (/reference/java/lang/String) `alias)` Allows this receiver to select the alias for a private k
`void`	`onComplianceAcknowledgementRequired` (/reference/android/app/admin/DeviceAdminReceive (/reference/android/content/Context) `context, I` Called to notify a profile owner of an organization-o maximum profile time off policy.
`CharSequence` (/reference/java/lang/CharSequence)	`onDisableRequested` (/reference/android/app/adr (/reference/android/content/Context) `context, I` Called when the user has asked to disable the admi (/reference/android/app/admin/DeviceAdminReceive
`void`	`onDisabled` (/reference/android/app/admin/Device `context, Intent` (/reference/android/content/Inte

	Called prior to the administrator being disabled, as a (/reference/android/app/admin/DeviceAdminReceive
`void`	onEnabled (/reference/android/app/admin/DeviceA `context, Intent` (/reference/android/content/Inte Called after the administrator is first enabled, as a re (/reference/android/app/admin/DeviceAdminReceive
`void`	onLockTaskModeEntering (/reference/android/app/admin/DeviceAdminReceive (/reference/android/content/Context) `context, I` Called when a device is entering lock task mode.
`void`	onLockTaskModeExiting (/reference/android/app (/reference/android/content/Context) `context, I` Called when a device is exiting lock task mode.
`void`	onNetworkLogsAvailable (/reference/android/ap (`Context` (/reference/android/content/Context) `co` Called each time a new batch of network logs can b
`void`	onOperationSafetyStateChanged (/reference/a (`Context` (/reference/android/content/Context) `co` Called to notify the state of operations that can be u
`void`	onPasswordChanged (/reference/android/app/admi (`Context` (/reference/android/content/Context) `co` Called after the user has changed their device or pr (/reference/android/app/admin/DeviceAdminReceive
`void`	onPasswordChanged (/reference/android/app/admi (/reference/android/content/Context) `context, I` *This method was deprecated in API level 26. From* Bu onPasswordChanged(android.content.Contex (/reference/android/app/admin/DeviceAdminReceive

void	onPasswordExpiring (/reference/android/app/adr (/reference/android/content/Context) context, I *This method was deprecated in API level 26. From* Bu onPasswordExpiring(android.content.Conte (/reference/android/app/admin/DeviceAdminReceive
void	onPasswordExpiring (/reference/android/app/adr (Context (/reference/android/content/Context) co Called periodically when the device or profile challer
void	onPasswordFailed (/reference/android/app/admin (Context (/reference/android/content/Context) co Called after the user has failed at entering their devi (/reference/android/app/admin/DeviceAdminReceive
void	onPasswordFailed (/reference/android/app/admin (/reference/android/content/Context) context, I *This method was deprecated in API level 26. From* Bu onPasswordFailed(android.content.Contex (/reference/android/app/admin/DeviceAdminReceive
void	onPasswordSucceeded (/reference/android/app/ac (/reference/android/content/Context) context, I *This method was deprecated in API level 26. From* Bu onPasswordSucceeded(android.content.Cont (/reference/android/app/admin/DeviceAdminReceive
void	onPasswordSucceeded (/reference/android/app/admin/DeviceAdminReceive (/reference/android/content/Context) context, I Called after the user has succeeded at entering thei (/reference/android/app/admin/DeviceAdminReceive
void	onProfileProvisioningComplete (/reference/a (Context (/reference/android/content/Context) co

	Called when provisioning of a managed profile or ma
void	onReadyForUserInitialization (/reference/an (Context (/reference/android/content/Context) co *This method was deprecated in API level 24. Do not*
void	onReceive (/reference/android/app/admin/DeviceA context, Intent (/reference/android/content/Inte Intercept standard device administrator broadcasts.
void	onSecurityLogsAvailable (/reference/android/a (/reference/android/content/Context) context, I Called when a new batch of security logs can be ret
void	onSystemUpdatePending (/reference/android/app (/reference/android/content/Context) context, I Called when the information about a pending system
void	onTransferAffiliatedProfileOwnershipCom (/reference/android/app/admin/DeviceAdminReceive (/reference/android/content/Context) context, U Called on the device owner when the ownership of c
void	onTransferOwnershipComplete (/reference/android/app/admin/DeviceAdminReceive (/reference/android/content/Context) context, P Called on the newly assigned owner (either device o
void	onUserAdded (/reference/android/app/admin/Devic (/reference/android/content/Context) context, I Called when a user or profile is created.
void	onUserRemoved (/reference/android/app/admin/De (/reference/android/content/Context) context, I

	Called when a user or profile is removed.
`void`	`onUserStarted` (/reference/android/app/admin/Dev (/reference/android/content/Context) `context, I` Called when a user or profile is started.
`void`	`onUserStopped` (/reference/android/app/admin/Dev (/reference/android/content/Context) `context, I` Called when a user or profile is stopped.
`void`	`onUserSwitched` (/reference/android/app/admin/D (/reference/android/content/Context) `context, I` Called when a user or profile is switched to.

Inherited methods

From class `android.content.BroadcastReceiver` (/reference/android/content/BroadcastReceiver

`final void`	`abortBroadcast` (/refe Sets the flag indicating `Context.sendOrdere` (/reference/android/cor
`final void`	`clearAbortBroadcas` Clears the flag indicatin
`final boolean`	`getAbortBroadcast` (Returns the flag indicati
`final boolean`	`getDebugUnregister` Return the last value giv (/reference/android/cor

`final int`	`getResultCode` (/refer Retrieve the current res
`final String` (/reference/java/lang/String)	`getResultData` (/refer Retrieve the current res
`final Bundle` (/reference/android/os/Bundle)	`getResultExtras` (/re Retrieve the current res
`String` (/reference/java/lang/String)	`getSentFromPackage` Returns the package na
`int`	`getSentFromUid` (/refe Returns the uid of the a
`final BroadcastReceiver.PendingResult` (/reference/android/content/BroadcastReceiver.PendingResult)	`goAsync` (/reference/an This can be called by an (/reference/android/cor to keep the broadcast a
`final boolean`	`isInitialStickyBro` Returns true if the recei broadcast and is curren
`final boolean`	`isOrderedBroadcast` Returns true if the recei
`abstract void`	`onReceive` (/reference/android/cor (/reference/android/cor This method is called wh

IBinder (/reference/android/os/IBinder)	peekService (/reference/android/cor (Context (/reference/a Provide a binder to an a
final void	setDebugUnregister Control inclusion of deb (/reference/android/cor .
final void	setOrderedHint (/refe For internal use, sets the
final void	setResult (/reference/ code, String (/refere Change all of the result Context.sendOrdere (/reference/android/cor
final void	setResultCode (/refer Change the current resu Context.sendOrdere (/reference/android/cor
final void	setResultData (/refer (/reference/java/lang/St Change the current resu Context.sendOrdere (/reference/android/cor
final void	setResultExtras (/re (/reference/android/os/I Change the current resu Context.sendOrdere (/reference/android/cor

From class `java.lang.Object` (/reference/java/lang/Object)

`Object` (/reference/java/lang/Object)	`clone` (/reference/java/lang/Object#clone())`()` Creates and returns a copy of this object.
`boolean`	`equals` (/reference/java/lang/Object#equals(java.lang.Ob Indicates whether some other object is "equal to" this one
`void`	`finalize` (/reference/java/lang/Object#finalize())`()` Called by the garbage collector on an object when garbag
`final Class` (/reference/java/lang/Class)`<?>`	`getClass` (/reference/java/lang/Object#getClass())`()` Returns the runtime class of this `Object`.
`int`	`hashCode` (/reference/java/lang/Object#hashCode())`()` Returns a hash code value for the object.
`final void`	`notify` (/reference/java/lang/Object#notify())`()` Wakes up a single thread that is waiting on this object's m
`final void`	`notifyAll` (/reference/java/lang/Object#notifyAll())`()` Wakes up all threads that are waiting on this object's mon
`String` (/reference/java/lang/String)	`toString` (/reference/java/lang/Object#toString())`()` Returns a string representation of the object.
`final void`	`wait` (/reference/java/lang/Object#wait(long,%20int))`(lo` Causes the current thread to wait until it is awakened, typ
`final void`	`wait` (/reference/java/lang/Object#wait(long))`(long tir` Causes the current thread to wait until it is awakened, typ

`final void`	`wait` (/reference/java/lang/Object#wait())`()` Causes the current thread to wait until it is awakened, typ

Constants

ACTION_CHOOSE_PRIVATE_KEY_ALIAS

Added in API level 29 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_CHOOSE_PRIVATE_KEY_ALIAS
```

Broadcast action: notify that some app is attempting to choose a KeyChain key.

See also:

`onChoosePrivateKeyAlias(Context, Intent, int, Uri, String)` (/reference/android/app/admin/DeviceAdminReceiver#onChoosePrivateKeyAlias(android.content.Context,%20android.content.Intent,%20int,%20android.net.Uri,%20java.lang.String))

`DelegatedAdminReceiver.onChoosePrivateKeyAlias(Context, Intent, int, Uri, String)` (/reference/android/app/admin/DelegatedAdminReceiver#onChoosePrivateKeyAlias(android.content.Context,%20android.content.Intent,%20int,%20android.net.Uri,%20java.lang.String))

Constant Value: "android.app.action.CHOOSE_PRIVATE_KEY_ALIAS"

ACTION_DEVICE_ADMIN_DISABLED

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_DEVICE_ADMIN_DISABLED
```

Action sent to a device administrator when the user has disabled it. Upon return, the application no longer has access to the protected device policy manager APIs. You will generally handle this in `DeviceAdminReceiver#onDisabled(Context, Intent)`

(/reference/android/app/admin/DeviceAdminReceiver#onDisabled(android.content.Context,%20android.content.Intent))
. Note that this action will be sent the receiver regardless of whether it is explicitly listed in its intent filter.

Constant Value: "android.app.action.DEVICE_ADMIN_DISABLED"

ACTION_DEVICE_ADMIN_DISABLE_REQUESTED

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_DEVICE_ADMIN_DISABLE_REQ
```

Action sent to a device administrator when the user has requested to disable it, but before this has actually been done. This gives you a chance to supply a message to the user about the impact of disabling your admin, by setting the extra field `EXTRA_DISABLE_WARNING` (/reference/android/app/admin/DeviceAdminReceiver#EXTRA_DISABLE_WARNING) in the result Intent. If not set, no warning will be displayed. If set, the given text will be shown to the user before they disable your admin.

Constant Value: "android.app.action.DEVICE_ADMIN_DISABLE_REQUESTED"

ACTION_DEVICE_ADMIN_ENABLED

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_DEVICE_ADMIN_ENABLED
```

This is the primary action that a device administrator must implement to be allowed to manage a device. This will be set to the receiver when the user enables it for administration. You will generally handle this in `DeviceAdminReceiver#onEnabled(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onEnabled(android.content.Context,%20android.content.Intent))
. To be supported, the receiver must also require the `Manifest.permission.BIND_DEVICE_ADMIN` (/reference/android/Manifest.permission#BIND_DEVICE_ADMIN) permission so that other applications can not abuse it.

Constant Value: "android.app.action.DEVICE_ADMIN_ENABLED"

ACTION_LOCK_TASK_ENTERING

Added in API level 21 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_LOCK_TASK_ENTERING
```

Action sent to a device administrator to notify that the device is entering lock task mode. The extra `EXTRA_LOCK_TASK_PACKAGE` (/reference/android/app/admin/DeviceAdminReceiver#EXTRA_LOCK_TASK_PACKAGE) will describe the package using lock task mode.

The calling device admin must be the device owner or profile owner to receive this broadcast.

See also:

`DevicePolicyManager.isLockTaskPermitted(String)` (/reference/android/app/admin/DevicePolicyManager#isLockTaskPermitted(java.lang.String))

Constant Value: "android.app.action.LOCK_TASK_ENTERING"

ACTION_LOCK_TASK_EXITING

Added in API level 21 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_LOCK_TASK_EXITING
```

Action sent to a device administrator to notify that the device is exiting lock task mode.

The calling device admin must be the device owner or profile owner to receive this broadcast.

See also:

`DevicePolicyManager.isLockTaskPermitted(String)` (/reference/android/app/admin/DevicePolicyManager#isLockTaskPermitted(java.lang.String))

Constant Value: "android.app.action.LOCK_TASK_EXITING"

ACTION_NETWORK_LOGS_AVAILABLE

Added in API level 29 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_NETWORK_LOGS_AVAILABLE
```

Broadcast action: notify that a new batch of network logs is ready to be collected.

See also:

`onNetworkLogsAvailable(Context, Intent, long, int)` (/reference/android/app/admin/DeviceAdminReceiver#onNetworkLogsAvailable(android.content.Context,%20android.content.Intent,%20long,%20int))

`DelegatedAdminReceiver.onNetworkLogsAvailable(Context, Intent, long, int)` (/reference/android/app/admin/DelegatedAdminReceiver#onNetworkLogsAvailable(android.content.Context,%20android.content.Intent,%20long,%20int))

Constant Value: "android.app.action.NETWORK_LOGS_AVAILABLE"

ACTION_PASSWORD_CHANGED

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_PASSWORD_CHANGED
```

Action sent to a device administrator when the user has changed the password of their device or profile challenge. You can at this point check the characteristics of the new password with `DevicePolicyManager.isActivePasswordSufficient()` (/reference/android/app/admin/DevicePolicyManager#isActivePasswordSufficient()). You will generally handle this in `DeviceAdminReceiver#onPasswordChanged(Context, Intent, UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordChanged(android.content.Context,%20android.content.Intent,%20android.os.UserHandle))
.

The calling device admin must have requested `DeviceAdminInfo#USES_POLICY_LIMIT_PASSWORD` (/reference/android/app/admin/DeviceAdminInfo#USES_POLICY_LIMIT_PASSWORD) to receive this broadcast.

Constant Value: "android.app.action.ACTION_PASSWORD_CHANGED"

ACTION_PASSWORD_EXPIRING

Added in API level 11 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_PASSWORD_EXPIRING
```

Action periodically sent to a device administrator when the device or profile challenge password is expiring. You will generally handle this in `DeviceAdminReceiver#onPasswordExpiring(Context, Intent, UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordExpiring(android.content.Context,%20android.content.Intent,%20android.os.UserHandle)).

The calling device admin must have requested `DeviceAdminInfo#USES_POLICY_EXPIRE_PASSWORD` (/reference/android/app/admin/DeviceAdminInfo#USES_POLICY_EXPIRE_PASSWORD) to receive this broadcast.

Constant Value: "android.app.action.ACTION_PASSWORD_EXPIRING"

ACTION_PASSWORD_FAILED

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_PASSWORD_FAILED
```

Action sent to a device administrator when the user has entered an incorrect device or profile challenge password. You can at this point check the number of failed password attempts there have been with `DevicePolicyManager.getCurrentFailedPasswordAttempts()` (/reference/android/app/admin/DevicePolicyManager#getCurrentFailedPasswordAttempts()). You will generally handle this in `DeviceAdminReceiver#onPasswordFailed(Context, Intent, UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordFailed(android.content.Context,%20android.content.Intent,%20android.os.UserHandle)).

The calling device admin must have requested `DeviceAdminInfo#USES_POLICY_WATCH_LOGIN`

(/reference/android/app/admin/DeviceAdminInfo#USES_POLICY_WATCH_LOGIN) to receive this broadcast.

Constant Value: "android.app.action.ACTION_PASSWORD_FAILED"

ACTION_PASSWORD_SUCCEEDED

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_PASSWORD_SUCCEEDED
```

Action sent to a device administrator when the user has successfully entered their device or profile challenge password, after failing one or more times. You will generally handle this in `DeviceAdminReceiver#onPasswordSucceeded(Context, Intent, UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordSucceeded(android.content.Context,%20android.content.Intent,%20android.os.UserHandle)).

The calling device admin must have requested `DeviceAdminInfo#USES_POLICY_WATCH_LOGIN` (/reference/android/app/admin/DeviceAdminInfo#USES_POLICY_WATCH_LOGIN) to receive this broadcast.

Constant Value: "android.app.action.ACTION_PASSWORD_SUCCEEDED"

ACTION_PROFILE_PROVISIONING_COMPLETE

Added in API level 21 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_PROFILE_PROVISIONING_COM
```

Broadcast Action: This broadcast is sent to indicate that provisioning of a managed profile or managed device has completed successfully.

The broadcast is limited to the profile that will be managed by the application that requested provisioning. In the device owner case the profile is the primary user. The broadcast will also be limited to the `DeviceAdminReceiver` (/reference/android/app/admin/DeviceAdminReceiver) component specified in the original intent or NFC bump that started the provisioning

process (see `DevicePolicyManager.ACTION_PROVISION_MANAGED_PROFILE` (/reference/android/app/admin/DevicePolicyManager#ACTION_PROVISION_MANAGED_PROFILE)).

A device admin application which listens to this intent can find out if the device was provisioned for the device owner or profile owner case by calling respectively `DevicePolicyManager.isDeviceOwnerApp(String)` (/reference/android/app/admin/DevicePolicyManager#isDeviceOwnerApp(java.lang.String)) and `DevicePolicyManager.isProfileOwnerApp(String)` (/reference/android/app/admin/DevicePolicyManager#isProfileOwnerApp(java.lang.String)). You will generally handle this in `DeviceAdminReceiver#onProfileProvisioningComplete` (/reference/android/app/admin/DeviceAdminReceiver#onProfileProvisioningComplete(android.content.Context,%20android.content.Intent))

.

See also:

`DevicePolicyManager.ACTION_PROVISIONING_SUCCESSFUL` (/reference/android/app/admin/DevicePolicyManager#ACTION_PROVISIONING_SUCCESSFUL)

Constant Value: "android.app.action.PROFILE_PROVISIONING_COMPLETE"

ACTION_SECURITY_LOGS_AVAILABLE

Added in API level 31 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) ACTION_SECURITY_LOGS_AVAILABLE
```

Broadcast action: notify that a new batch of security logs is ready to be collected.

Constant Value: "android.app.action.SECURITY_LOGS_AVAILABLE"

BUGREPORT_FAILURE_FAILED_COMPLETING

Added in API level 24 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final int BUGREPORT_FAILURE_FAILED_COMPLETING
```

Bugreport completion process failed.

If this error code is received, the requesting of bugreport can be retried.

See also:

`DevicePolicyManager.requestBugreport(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#requestBugreport(android.content.ComponentName))

Constant Value: 0 (0x00000000)

BUGREPORT_FAILURE_FILE_NO_LONGER_AVAILABLE

Added in API level 24 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final int BUGREPORT_FAILURE_FILE_NO_LONGER_AVAILABLE
```

Bugreport has been created, but is no longer available for collection.

This error likely occurs because the user of the device hasn't consented to share the bugreport for a long period after its creation.

If this error code is received, the requesting of bugreport can be retried.

See also:

`DevicePolicyManager.requestBugreport(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#requestBugreport(android.content.ComponentName))

Constant Value: 1 (0x00000001)

DEVICE_ADMIN_META_DATA

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) DEVICE_ADMIN_META_DATA
```

Name under which a DevicePolicy component publishes information about itself. This meta-data must reference an XML resource containing a device-admin tag.

Constant Value: "android.app.device_admin"

EXTRA_DISABLE_WARNING

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) EXTRA_DISABLE_WARNING
```

A CharSequence that can be shown to the user informing them of the impact of disabling your admin.

See also:

`ACTION_DEVICE_ADMIN_DISABLE_REQUESTED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ADMIN_DISABLE_REQUESTED)

Constant Value: "android.app.extra.DISABLE_WARNING"

EXTRA_LOCK_TASK_PACKAGE

Added in API level 21 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) EXTRA_LOCK_TASK_PACKAGE
```

A string containing the name of the package entering lock task mode.

See also:

`ACTION_LOCK_TASK_ENTERING` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_LOCK_TASK_ENTERING)

Constant Value: "android.app.extra.LOCK_TASK_PACKAGE"

EXTRA_TRANSFER_OWNERSHIP_ADMIN_EXTRAS_BUNDLE

Added in API level 28 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static final String (/reference/java/lang/String) EXTRA_TRANSFER_OWNERSHIP_ADMIN_
```

A `Parcelable` (/reference/android/os/Parcelable) extra of type `PersistableBundle` (/reference/android/os/PersistableBundle) that allows a mobile device management application to pass data to the management application instance after owner transfer.

If the transfer is successful, the new owner receives the data in `DeviceAdminReceiver#onTransferOwnershipComplete(Context, PersistableBundle)` (/reference/android/app/admin/DeviceAdminReceiver#onTransferOwnershipComplete(android.content.Context,%20android.os.PersistableBundle)). The bundle is not changed during the ownership transfer.

See also:

`DevicePolicyManager.transferOwnership(ComponentName, ComponentName, PersistableBundle)` (/reference/android/app/admin/DevicePolicyManager#transferOwnership(android.content.ComponentName,%20android.content.ComponentName,%20android.os.PersistableBundle))

Constant Value: "android.app.extra.TRANSFER_OWNERSHIP_ADMIN_EXTRAS_BUNDLE"

Public constructors

DeviceAdminReceiver

```
public DeviceAdminReceiver ()
```

Public methods

getManager

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public DevicePolicyManager (/reference/android/app/admin/DevicePolicyManager) getManager (
```

Retrieve the DevicePolicyManager interface for this administrator to work with the system.

Parameters

`context`	`Context`: This value cannot be `null`.

Returns

`DevicePolicyManager` (/reference/android/app/admin/DevicePolicyManager)	This value cannot be `null`.

getWho

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public ComponentName (/reference/android/content/ComponentName) getWho (Context (/reference
```

Retrieve the ComponentName describing who this device administrator is, for use in `DevicePolicyManager` (/reference/android/app/admin/DevicePolicyManager) APIs that require the administrator to identify itself.

Parameters

`context`	`Context`: This value cannot be `null`.

Returns

`ComponentName` (/reference/android/content/ComponentName)	This value cannot be `null`.

onBugreportFailed

Added in API level 24 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onBugreportFailed (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                int failureCode)
```

Called when the bugreport collection flow has failed.

This callback is only applicable to device owners.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`failureCode`	`int`: int containing failure code. One of `BUGREPORT_FAILURE_FAILED_` (/reference/android/app/admin/DeviceAdminReceiver#BUGREPORT_FAIL `FAILURE_FILE_NO_LONGER_AVAILABLE` (/reference/android/app/admin/DeviceAdminReceiver#BUGREPORT_FAIL `BUGREPORT_FAILURE_FAILED_COMPLETING` (/reference/android/app/admin/DeviceAdminReceiver#BUGREPORT_FAIL `FAILURE_FILE_NO_LONGER_AVAILABLE` (/reference/android/app/admin/DeviceAdminReceiver#BUGREPORT_FAIL

See also:

`DevicePolicyManager.requestBugreport(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#requestBugreport(android.content.ComponentNa me))

onBugreportShared

Added in API level 24 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onBugreportShared (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                String (/reference/java/lang/String) bugreportHash)
```

Called when the bugreport has been shared with the device administrator app.

This callback is only applicable to device owners.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . Contains the URI of the bugreport file (with MIME type "application/vnd. calling `Intent#getData()` (/reference/android/content/Intent#getData
`bugreportHash`	`String`: SHA-256 hash of the bugreport file. This value cannot be `null`.

See also:

`DevicePolicyManager.requestBugreport(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#requestBugreport(android.content.ComponentNa me))

onBugreportSharingDeclined

Added in API level 24 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onBugreportSharingDeclined (Context (/reference/android/content/Context) cont
                Intent (/reference/android/content/Intent) intent)
```

Called when sharing a bugreport has been cancelled by the user of the device.

This callback is only applicable to device owners.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.

See also:

`DevicePolicyManager.requestBugreport(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#requestBugreport(android.content.ComponentName))

onChoosePrivateKeyAlias

Added in API level 23 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public String (/reference/java/lang/String) onChoosePrivateKeyAlias (Context (/reference/and
                Intent (/reference/android/content/Intent) intent,
                int uid,
                Uri (/reference/android/net/Uri) uri,
                String (/reference/java/lang/String) alias)
```

Allows this receiver to select the alias for a private key and certificate pair for authentication. If this method returns null, the default `Activity` (/reference/android/app/Activity) will be shown that lets the user pick a private key and certificate pair. If this method returns `KeyChain#KEY_ALIAS_SELECTION_DENIED` (/reference/android/security/KeyChain#KEY_ALIAS_SELECTION_DENIED), the default `Activity`

(/reference/android/app/Activity) will not be shown and the user will not be allowed to pick anything. And the app, that called `KeyChain#choosePrivateKeyAlias` (/reference/android/security/KeyChain#choosePrivateKeyAlias(android.app.Activity,%20android.security.KeyChainAliasCallback,%20java.lang.String[],%20java.security.Principal[],%20android.net.Uri,%20java.lang.String)) , will receive `null` back.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`uid`	`int`: The uid of the app asking for the private key and certificate pair.
`uri`	`Uri`: The URI to authenticate, may be null.
`alias`	`String`: The alias preselected by the client, or null.

Returns

`String` (/reference/java/lang/String)	The private key alias to return and grant access to.

See also:

`KeyChain.choosePrivateKeyAlias(Activity, KeyChainAliasCallback, String, Principal, Uri, String)` (/reference/android/security/KeyChain#choosePrivateKeyAlias(android.app.Activity,%20android.security.KeyChainAliasCallback,%20java.lang.String[],%20java.security.Principal[],%20android.net.Uri,%20java.lang.String))

onComplianceAcknowledgementRequired

Added in API level 31 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onComplianceAcknowledgementRequired (Context (/reference/android/content/Co
                Intent (/reference/android/content/Intent) intent)
```

Called to notify a profile owner of an organization-owned device that it needs to acknowledge device compliance to allow the user to turn the profile off if needed according to the maximum profile time off policy. Default implementation acknowledges compliance immediately. DPC may prefer to override this implementation to delay acknowledgement until a successful policy sync. Until compliance is acknowledged the user is still free to turn the profile off, but the timer won't be reset, so personal apps will be suspended sooner. This callback is delivered using a foreground broadcast and should be handled quickly.

Parameters

`context`	`Context`: the running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi This value cannot be **`null`**.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.

See also:

`DevicePolicyManager.acknowledgeDeviceCompliant()` (/reference/android/app/admin/DevicePolicyManager#acknowledgeDeviceCompliant())

`DevicePolicyManager.isComplianceAcknowledgementRequired()` (/reference/android/app/admin/DevicePolicyManager#isComplianceAcknowledgementRequired())

`DevicePolicyManager.setManagedProfileMaximumTimeOff(ComponentName, long)` (/reference/android/app/admin/DevicePolicyManager#setManagedProfileMaximumTimeOff(android.content.ComponentName,%20long))

onDisableRequested

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public CharSequence (/reference/java/lang/CharSequence) onDisableRequested (Context (/refe
                Intent (/reference/android/content/Intent) intent)
```

Called when the user has asked to disable the administrator, as a result of receiving `ACTION_DEVICE_ADMIN_DISABLE_REQUESTED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ADMIN_DISABLE_REQUESTED), giving you a chance to present a warning message to them. The message is returned as the result; if null is returned (the default implementation), no message will be displayed.

Parameters

`context`	**`Context`**: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`intent`	**`Intent`**: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.

Returns

`CharSequence`	Return the warning message to display to the user before being disabled; if null is returned, no message is displayed.

(/reference/java/lang/CharSequence)

onDisabled

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onDisabled (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

Called prior to the administrator being disabled, as a result of receiving `ACTION_DEVICE_ADMIN_DISABLED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ADMIN_DISABLED). Upon return, you can no longer use the protected parts of the `DevicePolicyManager` (/reference/android/app/admin/DevicePolicyManager) API.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.

onEnabled

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onEnabled (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

Called after the administrator is first enabled, as a result of receiving `ACTION_DEVICE_ADMIN_ENABLED`

(/reference/android/app/admin/DeviceAdminReceiver#ACTION_DEVICE_ADMIN_ENABLED). At this point you can use `DevicePolicyManager` (/reference/android/app/admin/DevicePolicyManager) to set your desired policies.

If the admin is activated by a device owner, then the intent may contain private extras that are relevant to user setup.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.

See also:

`DevicePolicyManager.createAndManageUser(ComponentName, String, ComponentName, PersistableBundle, int)` (/reference/android/app/admin/DevicePolicyManager#createAndManageUser(android.content.ComponentName,%20java.lang.String,%20android.content.ComponentName,%20android.os.PersistableBundle,%20int))

onLockTaskModeEntering

Added in API level 21 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onLockTaskModeEntering (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                String (/reference/java/lang/String) pkg)
```

Called when a device is entering lock task mode.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`pkg`	`String`: The authorized package using lock task mode. This value canno

onLockTaskModeExiting

Added in API level 21 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onLockTaskModeExiting (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

Called when a device is exiting lock task mode.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.

onNetworkLogsAvailable

Added in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onNetworkLogsAvailable (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                long batchToken,
                int networkLogsCount)
```

Called each time a new batch of network logs can be retrieved. This callback method will only ever be called when network logging is enabled. The logs can only be retrieved while network logging is enabled.

If a secondary user or profile is created, this callback won't be received until all users become affiliated again (even if network logging is enabled). It will also no longer be possible to retrieve the network logs batch with the most recent `batchToken` provided by this callback. See `DevicePolicyManager#setAffiliationIds` (/reference/android/app/admin/DevicePolicyManager#setAffiliationIds(android.content.ComponentName, %20java.util.Set<java.lang.String>))
.

This callback is only applicable to device owners and profile owners.

This callback is triggered by a foreground broadcast and the app should ensure that any long-running work is not executed synchronously inside the callback.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`batchToken`	`long`: The token representing the current batch of network logs.

`networkLogsCount`	`int`: The total count of events in the current batch of network logs. Value

See also:

`DevicePolicyManager.retrieveNetworkLogs(ComponentName, long)` (/reference/android/app/admin/DevicePolicyManager#retrieveNetworkLogs(android.content.ComponentName,%20long))

onOperationSafetyStateChanged

Added in API level 31 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onOperationSafetyStateChanged (Context (/reference/android/content/Context) c
                int reason,
                boolean isSafe)
```

Called to notify the state of operations that can be unsafe to execute has changed.

Note:/b> notice that the operation safety state might change between the time this callback is received and the operation's method on `DevicePolicyManager` (/reference/android/app/admin/DevicePolicyManager) is called, so calls to the latter could still throw a `UnsafeStateException` (/reference/android/app/admin/UnsafeStateException) even when this method is called with `isSafe` as `true`

Parameters

`context`	`Context`: the running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi This value cannot be `null`.
`reason`	`int`: the reason an operation could be unsafe. Value is android.app.admin.DevicePolicyManager.OPERATION_SAFETY_REASON_ `OPERATION_SAFETY_REASON_DRIVING_DISTRACTION` (/reference/android/app/admin/DevicePolicyManager#OPERATION_SAFE

`isSafe`	**`boolean`**: whether the operation is safe to be executed.

onPasswordChanged

Added in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordChanged (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) user)
```

Called after the user has changed their device or profile challenge password, as a result of receiving `ACTION_PASSWORD_CHANGED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWORD_CHANGED). At this point you can use `DevicePolicyManager#getPasswordQuality(android.content.ComponentName)` (/reference/android/app/admin/DevicePolicyManager#getPasswordQuality(android.content.ComponentName)) to retrieve the active password characteristics.

Parameters

`context`	**`Context`**: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be **`null`**.
`intent`	**`Intent`**: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be **`null`**.

`user`	`UserHandle`: The user or profile for whom the password changed. To se parent user, check for equality with `Process#myUserHandle` (/referenc value cannot be `null`.

onPasswordChanged

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)
Deprecated in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordChanged (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

This method was deprecated in API level 26.

From `Build.VERSION_CODES.O` (/reference/android/os/Build.VERSION_CODES#O), use `onPasswordChanged(android.content.Context, android.content.Intent, android.os.UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordChanged(android.content.Context,%20android.content.Intent,%20android.os.UserHandle)) instead.

Called after the user has changed their device or profile challenge password, as a result of receiving `ACTION_PASSWORD_CHANGED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWORD_CHANGED). At this point you can use `DevicePolicyManager#getPasswordQuality(android.content.ComponentName)` (/reference/android/app/admin/DevicePolicyManager#getPasswordQuality(android.content.ComponentName)) to retrieve the active password characteristics.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.

`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.

onPasswordExpiring

Added in API level 11 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

Deprecated in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordExpiring (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

This method was deprecated in API level 26.

From `Build.VERSION_CODES.O` (/reference/android/os/Build.VERSION_CODES#O)**, use** `onPasswordExpiring(android.content.Context, android.content.Intent, android.os.UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordExpiring(android.content.Context,%20android.content.Intent,%20android.os.UserHandle)) **instead.**

Called periodically when the device or profile challenge password is about to expire or has expired. It will typically be called at these times: on device boot, once per day before the password expires, and at the time when the password expires.

If the password is not updated by the user, this method will continue to be called once per day until the password is changed or the device admin disables password expiration.

The admin will typically post a notification requesting the user to change their password in response to this call. The actual password expiration time can be obtained by calling `DevicePolicyManager#getPasswordExpiration(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#getPasswordExpiration(android.content.ComponentName))

The admin should be sure to take down any notifications it posted in response to this call when it receives `DeviceAdminReceiver#onPasswordChanged(Context, Intent)`

(/reference/android/app/admin/DeviceAdminReceiver#onPasswordChanged(android.content.Context,%20android.content.Intent))

.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be `null`.

onPasswordExpiring **Added in API level 26** (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordExpiring (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) user)
```

Called periodically when the device or profile challenge password is about to expire or has expired. It will typically be called at these times: on device boot, once per day before the password expires, and at the time when the password expires.

If the password is not updated by the user, this method will continue to be called once per day until the password is changed or the device admin disables password expiration.

The admin will typically post a notification requesting the user to change their password in response to this call. The actual password expiration time can be obtained by calling `DevicePolicyManager#getPasswordExpiration(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#getPasswordExpiration(android.content.ComponentName))

The admin should be sure to take down any notifications it posted in response to this call when it receives `DeviceAdminReceiver#onPasswordChanged(Context, Intent, UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordChanged(android.content.Context,%20android.content.Intent,%20android.os.UserHandle))

.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`user`	`UserHandle`: The user or profile for whom the password is expiring. To s parent user, check for equality with `Process#myUserHandle` (/referenc value cannot be `null`.

onPasswordFailed

Added in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordFailed (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) user)
```

Called after the user has failed at entering their device or profile challenge password, as a result of receiving `ACTION_PASSWORD_FAILED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWORD_FAILED). At this point you can use `DevicePolicyManager#getCurrentFailedPasswordAttempts()`

(/reference/android/app/admin/DevicePolicyManager#getCurrentFailedPasswordAttempts()) **to retrieve the number of failed password attempts.**

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`user`	`UserHandle`: The user or profile for whom the password check failed. To a parent user, check for equality with `Process#myUserHandle` (/referer This value cannot be `null`.

onPasswordFailed

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

Deprecated in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordFailed (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

This method was deprecated in API level 26.

From `Build.VERSION_CODES.O` (/reference/android/os/Build.VERSION_CODES#O), use `onPasswordFailed(android.content.Context, android.content.Intent, android.os.UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordFailed(android.content.Context,%20android.content.Intent,%20android.os.UserHandle)) instead.

Called after the user has failed at entering their device or profile challenge password, as a result of receiving `ACTION_PASSWORD_FAILED`

(/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWORD_FAILED)**. At this point you can use** `DevicePolicyManager#getCurrentFailedPasswordAttempts()` (/reference/android/app/admin/DevicePolicyManager#getCurrentFailedPasswordAttempts()) **to retrieve the number of failed password attempts.**

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be `null`.

onPasswordSucceeded

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

Deprecated in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordSucceeded (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

This method was deprecated in API level 26.

From `Build.VERSION_CODES.O` (/reference/android/os/Build.VERSION_CODES#O)**, use** `onPasswordSucceeded(android.content.Context, android.content.Intent, android.os.UserHandle)` (/reference/android/app/admin/DeviceAdminReceiver#onPasswordSucceeded(android.content.Context,%20android.content.Intent,%20android.os.UserHandle)) **instead.**

Called after the user has succeeded at entering their device or profile challenge password, as a result of receiving `ACTION_PASSWORD_SUCCEEDED`

(/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWORD_SUCCEEDED). **This will only be received the first time they succeed after having previously failed.**

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be `null`.

onPasswordSucceeded

Added in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onPasswordSucceeded (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) user)
```

Called after the user has succeeded at entering their device or profile challenge password, as a result of receiving `ACTION_PASSWORD_SUCCEEDED` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PASSWORD_SUCCEEDED). This will only be received the first time they succeed after having previously failed.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi. This value cannot be `null`.

`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`user`	`UserHandle`: The user of profile for whom the password check succeed profile or a parent user, check for equality with `Process#myUserHandl` (/reference/android/os/Process#myUserHandle()). This value cannot be

onProfileProvisioningComplete

Added in API level 21 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onProfileProvisioningComplete (Context (/reference/android/content/Context) c
                Intent (/reference/android/content/Intent) intent)
```

Called when provisioning of a managed profile or managed device has completed successfully.

As a prerequisite for the execution of this callback the `DeviceAdminReceiver` (/reference/android/app/admin/DeviceAdminReceiver) has to declare an intent filter for `ACTION_PROFILE_PROVISIONING_COMPLETE` (/reference/android/app/admin/DeviceAdminReceiver#ACTION_PROFILE_PROVISIONING_COMPLETE). Its component must also be specified in the `DevicePolicyManager#EXTRA_DEVICE_ADMIN` (/reference/android/app/admin/DevicePolicyManager#EXTRA_DEVICE_ADMIN) of the `DevicePolicyManager#ACTION_PROVISION_MANAGED_PROFILE` (/reference/android/app/admin/DevicePolicyManager#ACTION_PROVISION_MANAGED_PROFILE) intent that started the managed provisioning.

When provisioning of a managed profile is complete, the managed profile is hidden until the profile owner calls `DevicePolicyManager#setProfileEnabled(ComponentName admin)` (/reference/android/app/admin/DevicePolicyManager#setProfileEnabled(android.content.ComponentNam e)) . Typically a profile owner will enable the profile when it has finished any additional setup such as adding an account by using the `AccountManager`

(/reference/android/accounts/AccountManager) **and calling APIs to bring the profile into the desired state.**

Note that provisioning completes without waiting for any server interactions, so the profile owner needs to wait for data to be available if required (e.g. Android device IDs or other data that is set as a result of server interactions).

From version `Build.VERSION_CODES.O` (/reference/android/os/Build.VERSION_CODES#O), when managed provisioning has completed, along with this callback the activity intent `DevicePolicyManager#ACTION_PROVISIONING_SUCCESSFUL` (/reference/android/app/admin/DevicePolicyManager#ACTION_PROVISIONING_SUCCESSFUL) will also be sent to the same application.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.

onReadyForUserInitialization

Added in **API level 23** (/guide/topics/manifest/uses-sdk-element#ApiLevels)
Deprecated in API level 24 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onReadyForUserInitialization (Context (/reference/android/content/Context) co
                Intent (/reference/android/content/Intent) intent)
```

This method was deprecated in API level 24.
Do not use

Called during provisioning of a managed device to allow the device initializer to perform user setup steps.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.

onReceive

Added in API level 8 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onReceive (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent)
```

Intercept standard device administrator broadcasts. Implementations should not override this method; it is better to implement the convenience callbacks for each action.

Parameters

`context`	`Context`: This value cannot be `null`.
`intent`	`Intent`: This value cannot be `null`.

onSecurityLogsAvailable Added in API level 24 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onSecurityLogsAvailable (Context (/reference/android/content/Context) context
                Intent (/reference/android/content/Intent) intent)
```

Called when a new batch of security logs can be retrieved.

If a secondary user or profile is created, this callback won't be received until all users become affiliated again (even if security logging is enabled). See `DevicePolicyManager#setAffiliationIds` (/reference/android/app/admin/DevicePolicyManager#setAffiliationIds(android.content.ComponentName, %20java.util.Set<java.lang.String>))

This callback will be re-triggered if the logs are not retrieved.

This callback is only applicable to device owners and profile owners of organization-owned managed profiles.

This callback is triggered by a foreground broadcast and the app should ensure that any long-running work is not executed synchronously inside the callback.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.

See also:

`DevicePolicyManager.retrieveSecurityLogs(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#retrieveSecurityLogs(android.content.ComponentName))

onSystemUpdatePending

Added in API level 23 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onSystemUpdatePending (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                long receivedTime)
```

Called when the information about a pending system update is available.

Allows the receiver to be notified when information about a pending system update is available from the system update service. The same pending system update can trigger multiple calls to this method, so it is necessary to examine the incoming parameters for details about the update.

This callback is only applicable to device owners and profile owners.

To get further information about a pending system update (for example, whether or not the update is a security patch), the device owner or profile owner can call `DevicePolicyManager#getPendingSystemUpdate` (/reference/android/app/admin/DevicePolicyManager#getPendingSystemUpdate(android.content.ComponentName))

.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.

`receivedTime`	`long`: The time as given by `System#currentTimeMillis()` (/referenc indicating when the current pending update was first available. -1 if no pe

See also:

`DevicePolicyManager.getPendingSystemUpdate(ComponentName)` (/reference/android/app/admin/DevicePolicyManager#getPendingSystemUpdate(android.content.Compo nentName))

onTransferAffiliatedProfileOwnershipComplete

Added in API level 28 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onTransferAffiliatedProfileOwnershipComplete (Context (/reference/android
            UserHandle (/reference/android/os/UserHandle) user)
```

Called on the device owner when the ownership of one of its affiliated profiles is transferred.

This can be used when transferring both device and profile ownership when using work profile on a fully managed device. The process would look like this:

1. **Transfer profile ownership**
2. **The device owner gets notified with this callback**
3. **Transfer device ownership**
4. **Both profile and device ownerships have been transferred**

Parameters

`context`	`Context`: the running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi This value cannot be `null`.

`user`	`UserHandle`: the `UserHandle` (/reference/android/os/UserHandle) of th

See also:

`DevicePolicyManager.transferOwnership(ComponentName, ComponentName, PersistableBundle)` (/reference/android/app/admin/DevicePolicyManager#transferOwnership(android.content.ComponentName,%20android.content.ComponentName,%20android.os.PersistableBundle))

onTransferOwnershipComplete

Added in API level 28 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onTransferOwnershipComplete (Context (/reference/android/content/Context) cor
                PersistableBundle (/reference/android/os/PersistableBundle) bundle)
```

Called on the newly assigned owner (either device owner or profile owner) when the ownership transfer has completed successfully.

The `bundle` parameter allows the original owner to pass data to the new one.

Parameters

`context`	`Context`: the running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi This value cannot be `null`.
`bundle`	`PersistableBundle`: the data to be passed to the new owner This valu

onUserAdded

Added in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onUserAdded (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) addedUser)
```

Called when a user or profile is created.

This callback is only applicable to device owners.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`addedUser`	`UserHandle`: The `UserHandle` (/reference/android/os/UserHandle) of th cannot be `null`.

onUserRemoved

Added in API level 26 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onUserRemoved (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) removedUser)
```

Called when a user or profile is removed.

This callback is only applicable to device owners.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`removedUser`	`UserHandle`: The `UserHandle` (/reference/android/os/UserHandle) of th cannot be `null`.

onUserStarted

Added in API level 28 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onUserStarted (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) startedUser)
```

Called when a user or profile is started.

This callback is only applicable to device owners.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be `null`.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi

. This value cannot be **null**.

`startedUser`	**UserHandle**: The **UserHandle** (/reference/android/os/UserHandle) of th cannot be **null**.

onUserStopped

Added in API level 28 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onUserStopped (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) stoppedUser)
```

Called when a user or profile is stopped.

This callback is only applicable to device owners.

Parameters

`context`	**Context**: The running context as per **onReceive(Context, Intent)** (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **null**.
`intent`	**Intent**: The received intent as per **onReceive(Context, Intent)** (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **null**.
`stoppedUser`	**UserHandle**: The **UserHandle** (/reference/android/os/UserHandle) of th cannot be **null**.

onUserSwitched

Added in API level 28 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void onUserSwitched (Context (/reference/android/content/Context) context,
                Intent (/reference/android/content/Intent) intent,
                UserHandle (/reference/android/os/UserHandle) switchedUser)
```

Called when a user or profile is switched to.

This callback is only applicable to device owners.

Parameters

`context`	`Context`: The running context as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`intent`	`Intent`: The received intent as per `onReceive(Context, Intent)` (/reference/android/app/admin/DeviceAdminReceiver#onReceive(androi . This value cannot be **`null`**.
`switchedUser`	`UserHandle`: The `UserHandle` (/reference/android/os/UserHandle) of tl value cannot be **`null`**.