

requiring printing and installing printing services that handle print requests. The print manager provides shared services with a UI system for printing, giving users consistent control over printing from any app. The print manager also ensures the security of content as it is passed through processing, from an app to a print service. You can add print support to your apps or develop print services to support specific types of printing. Printer manufacturers can use new APIs to develop their own print service – plug-in components that add vendor-specific logic and services to existing print services. They can build print and distribute them to Google Play, making it easy for users to find and install them on their devices. Similar to other apps, you can update your on-the-air print service at any time. Client apps can use new APIs to add printing capabilities to their apps with minimal code changes. In most cases, you would add a print action in your Action Bar and a UI for selecting the printed items. You would also implement APIs to create print jobs, query the manager for status, and cancel jobs. This allows you to print almost any type of content, from locally stored and documents to network data or a rendering view of a fabric. For wider compatibility, Android uses PDF as its main file format for before printing, your app needs to be a PDF viewer is well-painted in your content. For convenience, the print API provides native classes and WebViews to help you create PDFs using standard Android drawing APIs. If your app knows how to draw the content, it can quickly create a PDF to print. Most devices running Android 4.4 will include Google Cloud Printer pre-installed as a print service, as well as several Google apps that support printing, including Chrome, Drive, Gallery, and Quickoffice. Storage Access Boards A new storage accessibility makes it simple for users to browse and open documents, images, and other files across all the preferred document providers. A standard, easy-to-use UI allows users to browse files and access updates in a consistent manner across apps and providers. Box and others have integrated their services into the Storage Access Foundation, giving users easy access to documents from apps across the system. Cloud or on-premises storage services can participate in this ecosystem by applying a new document provider class that changes services. The provider class includes all the APIs needed to register the provider and the system and manage navigation, reading, and writing documents to the provider. The document provider can give users access to any remote or local data that can be represented as files – from text, pictures, and video wallpapers, audio, and more. If you build a document provider for a cloud or local service, you can deliver it to users as part of your existing app or app. After downloading and installing the app, users will have instant access to your service from any app that participates in the foundation. This can help you gain exposure and user engagement since users will find your services easier. If you develop a client app that manages files or documents, you can integrate with the Storage Access Foundation just by using the new CREATE_DOCUMENT or OPEN_DOCUMENT to open or create folders – the system automatically displays the standard UI for document navigation, including all available document providers. You can integrate your customer app one time, for all providers, without any vendor-specific code. As users add or remove their provider, they will continue to access their preferred service in your app, without changes or updates needed in your code. The storage foundation is integrated with the intention GET_CONTENT existing content, so users have also access to all their previous content and data sources from the new system UI for browsing. Apps can continue using a GET_CONTENT way to let users import data. Access to access storage and UI systems for navigation makes it easier for users to find and import the data from a wider range of sources. Most devices running Android 4.4 will include Google Drive and local storage pre-integrated as document providers, and Google which works with files also uses the new foundation. Low-power Detector Batch Android 4.4 introduces platform support for batch hardware, a new optimization that can dramatically reduce power consumption by continuous sensors activity. With Batch Sensors, Android works with hardware devices to collect and process event sensors efficiently in batch, rather than individually as detected. This allows the application process to remain in a low-power idle state until batch is delivered. You can request batch events from any sensors by using a standard event listener, and you can control the interval at which you receive baskets. You can also request immediate delivery of events between batch cycles. Batch sensor is ideal for low-power, long-running use-cases such as physical conditions, tracking locations, monitoring, and more. It can make your app more efficient and it allows you to track event sensors continuously – even while the screen is off and the system is sleeping. Sensor Batch is now available on Nexus 5, and we are working with our chipset partners to bring it to more devices as soon as possible. Moving and Runtastic Pedometer is using the step-detector piece to offer long-running, low-power services. Step Detector and Step Counter Android 4.4 also adds platform support for two new composite sensors – step sensors and step counters – which let your app step track when the user is walking, running, or climbing stairs. These new sensors are applied to low power consumption. Detect the analytical step to input accelerometer to recognize when the user took a step, then trigger an event with each step. The counter step tracks the total number of steps since the last device reboot and triggered an event with each change in the step count. Because the management logical and detector is built into the platform and underlying hardware, you don't need to maintain your own detection algorithms in your app. Step sensors and counter sensors are available on Nexus 5, and we are working with our chipset partners to bring them to new devices as soon as possible. SMS provider If you develop a messaging app that uses SMS or MMS, you can now use an SMS provider shared with new APIs to manage your app's message storage and retrieval. The new SMS provider and APIs define a standard interaction model for all apps that handle SMS or MMS messages. Along with the new provider and APIs, Android 4.4 introduces the new semantic to receive messages and postings to the provider. When receiving a message, the system routes it directly to the user's default app using the new SMS_DELIVER intent. Other apps can always listen for incoming messages using intents SMS_RECEIVED. Also, the system currently allows only the default app to write message data to the provider, although other apps can be read at any time. Contents Apps by default the user can still send messages – the system handles writing these messages to the provider on behalf of the app, so that users can see them in the default app. The new and semantic provider helps improve the user experience when several messaging apps are installed, and they help you build new messaging features with fully-supported, forward-compatible APIs. New ways to build beautiful apps A new immersive mode allows apps to use each pixel on the screen to display content and capture touch events. Full-screen Immersive Mode Now apps you can use each pixel on the device screen to display your content and capture touch events. Android 4.4 adds a new full-screen immersive mode that lets you create full-azure UIs reaching out of corner edge on phones and tablets, hiding all UI systems such as the status bar and navigation bar. It's ideal for rich content such as photos, videos, maps, books, and games. In the new mode, the system UI remains hidden, even while users interact with your app or game – you can capture touch events from anywhere across the screen, even areas that would otherwise be handled by giving them systems. This gives you a good way to create a larger, rich, more immersive UI in your app or game and also reduce visual distractions. To make sure that users always have easy, consistent access to UI systems from full-screen full-screen mode, Android 4.4 supports a new gesture – in immersive mode, a edge swipe from the top or bottom of the screen to now reveal the system UI. To return to immersive mode, users can handle the screen outside of the bar bound or wait for a short period for the auto-hiding bars. For a consistent user experience, the new gesture also works with previous methods for hiding the status bar. Transition Foundation for scenes animating Most apps structures around several UI key states that expose different actions. Many apps also use animation to help users understand the progress across these states and actions that are available to each. To make it easier to create high-quality animations in your app, Android 4.4 introduces a new Transition Foundation. The transition boards allow you to define scenes, typically view hierarchy, and transitions, which describe how to animate or transform the scenes when the user enters or exits them. You can use several predefined transition types to animate your scenes based on specific properties, such as layout bind, or visibility. There's also an auto-transition type that automatically limps, moves, and resizes input during a scene change. In addition, you can define custom transitions that hosted the properties that are problems in your app, and you can play into your own animation styles if necessary. With the frame transitions you can also animate changes to your UI on the fly, without needing to define scenes. For example, you can set a series of changes to a view hierarchy and then have the transitionManager to automatically run a transition delay on the changes. Once you've set up transitions, it's right to invoke them from your app. For example, you can call a single method to start a transition, make various changes in your view hierarchy, and on the next frame animation automatically starts that animated the changes you specified. Apps can use new window styles to request translucent system bars. For custom controls on the transitions that run between specific scenes in the flow of your application, you can use the TransitionManager. The TransitionManager allows you to define the relationship between the scenes and the transitions that run for specific scene changes. The translucent UI system that will estimate to find the most impact to your content, you can now use new window styles and themes to request translucent UI systems, including both the status bar and navigation bar. To ensure the credential of navigation bar buttons or status bar information, subtle gradients are shown behind the system bars. A typical use-case should be an app that needs to be shown in a wallpaper. Increased notification access notification service listeners can now see more information about incoming notifications that have been constructed using the built notification APIs. Service listeners can access a notification's actions as well as new extra fields – text, icon, pictures, progress, timer, and many others – extract cleaner information about the notification and present the information in a different way. Chromium WebView Android 4.4 includes a completely new application of WebView-based Chromium. The new WebView chromium gives you the latest in standard support, performance, and compatibility to build and display your web-based content. Chromium WebView provides great support for HTML5, CSS3, and JavaScript. It supports most of the HTML5 features available in Chrome for Android 3.0. It also brings an updated version of the JavaScript Engine (V8) that delivers dramatically improved JavaScript performance. Additionally, new Chromium WebView supports remote debugging using Chrome DevTools. For example, you can use Chrome DevTools on your development machine to inspect, debug, and analyze webView content you live on a mobile device. The new WebView chromium is included on all compatible devices running Android 4.4 and higher. You can take advantage of the new WebView right away, with minimal modification of existing apps and content. In most cases, your content will migrate to the new application without issue. Screen recording Now it's easy to create your high-quality video app, directly from your Android device. Android 4.4 adds support for screen recording and provides a screen recording utility that lets you start and stop recording on a connected device Settings your Android SDK on USB. It's a good new way to create walkthroughs and tutorials for your app, testing materials, marketing videos, and more. With the screen recording utility, you can capture video from your device's screen and store the video as an MP4 file on the device. You can record at any device resolution that is supported with bitrate you want, and the output holds the aspect ratio of the display. By default, the utility selects an equal or close resolution of the device's display resolution to the current orientation. When you are doing recordings, you can share the video directly from your device or zoom the MP4 file to your host computer for post-production. If your app plays videos or other protected content that you don't want to be captured by the screen, you can use SurfaceView.setSecure() to mark the content as secure. You can access screen recording through the adb tool included in the Android SDK, using the screen adb screen adb screen. You can also launch it in the Android Studio layout. Resolution required to switch to Android adaptive playback 4.4 brings formal support for playback adaptation to the Android media foundation. Adaptive playback is an optional feature to decode video for MPEG-DASH and other formats that enable meaningful changes to resolution during playback. The customer can start feeding the input video boards decode to a new resolution with the resolution of the output purchasers automatically, and without a space difference. Resolution required switching to Android 4.4 allows media apps to offer a significantly better streaming video experience. Apps can check for adaptive playback support to run using existing APIs and apply resolution-changing using new APIs introduced in Android 4.4. Common encryption for DASH Android now supports the Common Encryption (CENC) for MPEG-DASH, providing a standard, multiplatform DRM scheme for managing protected content. Apps can take advantage of CENC in Android's module FRAMEWORK_DRM and platform APIs to support DASH. HTTP Live Streaming Android 4.4 Updates on the HTTP Live PLATfOm (HLS) is supported in a subset of version 7 of the HLS specification (version 4 of the protocol). See the IETF Draft for details. Audio Tunneling in DSP For high-performance, lower-power audio playback, Android 4.4 adds platform support for tunny audio to a digital signal processor (DSP) of the device chips. With tunnels, audio decoder and production effects are cut-changed in the DSP, waking up the application processor less frequently and using less battery. Audio tunnels can dramatically improve battery life for use cases such as hearing music on a headset and the screen off. For example, with assorted audio, Nexus 5 offers a total time audio-network audio that reaches up to 60 hours, an increase of more than 50% on non-lined audio. Applications for the media take advantage of audio tunnels on supported devices without needing to modify code. The system applies tunneling to optimize audio playback whenever it is available on the device. The visualization of how the LoudnessEnhancer effect can make reversal content more audiences. Audio tunnels require support in hardware devices. Currently audio tunnels are available on Nexus 5 and we are working with our chipset partners to make it available on more devices as soon as possible. Audio Monitoring Apps can use new monitoring tools in the Visualizer effect to get updates on the peak and RMS level of any currently playing audio on the device. For example, you could use this certification to visualize music or to apply playback masters to a media player. Improved media playback applications can increase the content madress spoken by using the new LoudnessEnhancer effect, which acts as compressor and constant time that is specifically tuned for speech. The audio foundation can now report presentation timestamps from audio output to HAL to applications, for better audio-video synchronization. Audio timestamps let your app determine when a specific audio frame will (or was) introduce off-device to the user, you can use the timestamp information to more accurately synchronize audio and video frames. Wi-Fi Certified Miracast™ Android 4.4 devices can now be certified in the Wi-Fi Alliance Wi-Fi Display specifications as Miracast compatible. For help with testing, a new Wireless Display Developer option exposes advanced configuration controls and settings for Wireless Display certifications. You can access the Options in Settings > Developer Options > Wireless Display Certification. Nexus 5 is a certified Miracast wireless display device. RenderScript Compute Performance reference for Android 4.4 relative to Android 4.3, running on the same devices (Nexus 7, Nexus 10). Continuous performance improvements When your apps use RenderScript, they will benefit from continuous performance continuous performance of the RenderScript running itself, without the need for recompilation. The right chat shows profit performance of Android 4.4 on two popular chipsets. GPU Acceleration Any app using RenderScript on a device supports benefits from GPU acceleration, without code changes or recompile. Since the Nexus 10 first debuted RenderScript GPU acceleration, various other primary partners have added support. Now, with Android 4.4, GPU acceleration is available on the Nexus 5, as well as the Nexus 4, Nexus 7 (2013), and Nexus 10, and we are working with our partners to bring it to more devices as soon as possible. RenderScript to Android NDK now you can take advantage of RenderScript directly from your native code. A new C++ API in the Native Development Kit (NDK) allows you to access the same RenderScript function available through the APIs Foundation, including intrinsics, custom kernels, and more. If you have large, performance-intensive tasks handled in native code, you can perform these tasks using RenderScript and integrate them with your native code. RenderScript offers great performance across a wide range of devices, with automated support for multi-core CPUs, GPUs, and other processors. When you build an app that uses the RenderScript in the NDK, you can distribute it to any device running Android 2.2 or higher, just like with the RenderScript library available for APIs Foundation. Graphics GLES2.0 SurfaceFlinger Android 4.4 Upgrades its SurfaceFlinger from OpenGL ES1.0 to OpenGL ES 2.0. New Hardware composer support for displaying virtual version of Android Hardware Composer, HWComposer1.3, supports hardware composition of a virtual display in addition to the main, external (e.g. HDMI) display, and has improved OpenGL ES interoperability. New Type Connectivity New Bluetooth profiles Android 4.4 support for two new Bluetooth profiles to let apps support a wider range of low-power interactions and media. The Bluetooth HID on GATT (HOGP) gives apps a low-latency link with low-power peripheral devices such as muscleds, joysticks, and keyboards. Bluetooth MAP allows your apps message exchange with a nearby device, for example an automotive terminal for using mestrise or another mobile device. As a Bluetooth AVRCP 1.3 extension, users can now place absolute volume on the system from their Bluetooth device. Support platform for HOGP, MAP, and AVRCP built on Bluetooth stack presented by Google and Broadcom in Android 4.2. Support is available right away on Nexus devices and other Android-compatible devices that offer Bluetooth compatible capabilities. UR Blasters Android 4.4 introduces platform support for built-in IR explosions, along with a new API and system services that let you create APIs to take advantage of them. By using the new API, you can build apps that let users control nearby tv, tuners, switches, and other electronic devices. The API allows your app check if the phone or tablet has an infrared issuing, query it to insurance company frequency, and then sends infrared signals. Because the API is standard across Android devices running Android 4.4 or higher, your app can support the wider possible range of vendors without writing custom integration code. Wi-Fi TDLS supports Android 4.4 introduces a seamless way to current media and other data faster between devices already on the same Wi-Fi network by supporting Wi-Fi Titled Direct Installation Link (TDLS). Access system-wide settings for caption lock Android 4.4 now support a better access experience across apps by adding system-wide preferences for closed caption. Users can go to Settings > Accessibility > Captions to set global captioning preferences, as if shown and that language, text size, and text style to use. Apps that use video can now access the user's captioning settings and adjust presentations to meet the user's preferences. A new captioning manager API lets you check and control the user's captioning preferences. The caption manager provides you with the user's preferred captioning state as well as preferred local, factor bridge, and text style. The text style includes foreground and background color, edge properties, and typeface. Apps can now refer to the user-wide system preference. An example of the display system expected to display is shown right in the settings. In addition, apps that use VideoView can use a new API to declare that a stream caption along with a video stream to render. The system automatically handles the display of the captions on video frames according to the user's system settings. Currently, VideoView supports auto-display in captions in WebVTT format only. All apps showing captions should make sure to check the user's system captioning preferences and rendering captions as well as possible to those preferences. For more insight into how specific combinations of settings should look, you can look at a preview of captions in different languages, sizes, and styles right in the Settings app. Improved Access ANDROID APIs 4.4 extend the access APIs to support more structural and semantic description and observation of expressional components. With the NEW APIs, developers can improve the quality of accessible feedback by providing accessible services and more information about screen components. Through new access, developers can now determine if a noise is a popup, find its input type, and more. You can also use new APIs to work with roses that contain gis-like information, such as lists and tables. For example, you can now specify new supported actions, collection information, live region nodes, and more. New access events let developers better track the changes that are taking place in window content, and can now listen for changes in the handling exploration mode on the device. Support for International User Drawable Mirror for local RTL If your app is targeting users who use RTL scripts, you can use a new API to declare that a drawer should be auto-mirrored when the user's local settings include a RTL language. Declaring a drawn as auto-mirrored helps you prevent copies of advantages to your app and reduce the size of your APK. When you have drawable that is the reuse for both LTR and RTL presentations, you can declare the default versions as auto-mirrored and then auto-reflect those Drawables from your RTL resources. The Force RTL Layout option makes it easier to test your user's localization. You can declare various drawable types as auto-reflected in your application code, such as bitmap, layers, state listings, and other drawable. You can also declare a drawn as auto-reflected in your resource file by using a new attribute. Force RTL layout to make it easier to test and debug mirror layout problems without switching to a RTL language. Android includes a new developer option forced towards RTL layout in all apps. The Force RTL Layout option changes the device to RTL layout for all premises and displays text in your current language. This can help you find layout issues across your app, without having to display the app in a RTL language. You can access the options in Settings > Developer Options > layout RTL layout directives. SECURITY IMPROVEMENT SELinux (enhanced mode) Android 4.4 Updates its SELinux configuration from permissive in boot. This means potential violations are in a SELinux domain that a reinforced policy will block. Improving Android cryptographic algorithms has improved its security further by adding support for two more cryptographic algorithms. Elliptic Curve Digital Algorithm (ECDSA) support has been added to the keystore provider to enhance the security of digital signing, applicable to scenarios such as signing an application or a data connection. The key derivation derivation function is applied to protect the cryptographic keys used for full disk encryption. Other improvements on multizer devices, VPNS are now applied per app. This can allow a user to route all network traffic to a VPN without affecting other users on the device. Also, Android now supports FORTIFY_SOURCE level 2, and all codes compile with these protection. FORTIFY_SOURCE increasing to work with clang. Proccats A new tool called Proccats to help you analyze the resources your app's memory resources use, as well as the resources used by other apps and services running on the system. Proccats keep track of how apps are run on time, providing data on their execution duration and memory usage to help determine how their efficiency is performing. This is more important for apps that start services that run in the background, since it allows you to control how long they are running and how much RAM they are using while doing so. Proccats will also collect data for foreground applications over memory usage over time to determine the overall memory profile of the app. Proccats can help you identify background services started by your app. You can keep track of how long these services continue to run and how much RAM is used while doing so. Proccats also allow you your app's profile while it's in the foreground, using its memory usage over time to determine its overall memory profile. The new prostate tool lets you view details about memory usage for an app. You can access the proccats from the adb tool included in the SDK, but cut proccats.sterila. Also, for on-device device profiles, see the Process Developer Stats option, below. On-device memory status and profiling Android 4.4 includes a new developer option to make it easier to analyze your app's memory profile while running on a device or emulator. It's especially useful to get a view of how your app uses memory and performs on devices with low RAM. You can access the Options in Settings > Developer Options > Process Stats Process is a convenient way to check your app's memory usage. You can see how your app compares with other apps and zoom in on specific data about your app or it's background service. The Process Stats option shows you a variety of high-level guys on your app's memory usage, based on data collected using the new service proccats. On the main screen you can view a summary of memory system status. Green indicates the amount of time spent with low RAM usage, yellow indicates moderate ram usage, and red indicates high (critical) RAM usage below the summary is a summary list of each app's payload on the system. For each app, an Azure bar indicates the relative memory load completed (run x_avg_pss) in its processing, and a percentage number indicates the relative amount of time spent in the background. You can filter the list to display only foreground, background, or cash processes, and you can include or exclude system processes. You can also change the duration of the data collected from 3, 6, 12, or 24 hours, and you can include or exclude its memory. You can include, if your app is running longer than it should be or using too much memory over a period of time, there might be bugs in your code that you can resolve to improve your app's performance, especially when running on a low RAM device. RAM.

android studio gradle dependencies location , laser and feelings , terrarium tv latest apk , whatsapp plus apk 2020 iphone , fundamentals of vibrations meirovich solution manual , mapanej.pdf , automotive engineering fundamentals richard stone.pdf , 58273195072.pdf , family trust deed sample.pdf , the_ghost_bride_full_movie_2020.pdf , 5668269060.pdf , which earth material has the greatest permeability , normal_5f9516e224c4e.pdf , 3190264592.pdf .