I'm not robot
reCAPTCHA
Continue

Android studio preferences button

This topic describes some of the most commonly used preferences components and attributes that are used to build the settings screen. Preferred components This section describes common components of preferences. For more information, see the relevant reference pages for each component. PreferenceFragmentCompat - a fragment that handles the display of an interactive hierarchy of preferential objects. PreferenceScreen Preset Containers - A top-level container that represents the settings screen. This is the root part of the preferences hierarchy. PreferenceCategory - a container used to group similar preferences. PreferenceCategory displays the category name and visually separates groups of preferences. Individual Preference Preferences - the basic imaginable block, which represents the individual setting. If the preset is set to persist, it has a corresponding pair of keys and values that holds the user's choice for settings that can be accessed elsewhere in the application. EditTextPreference - A preference that persists in a string value. Users can tap preferences to start a dialog box that contains a text box that allows the user to change the permanent value. ListPreference - A preference that persists in a string value. Users can change this value in a dialog box that contains a list of radio buttons with corresponding labels. MultiSelectListPreference - A preference that persists in a set of strings. Users can change these values in a dialog box that contains a list of check boxes with corresponding labels. SeekBarPreference - A preference that persists with an integer value. You can change this value by dragging the appropriate lookup bar that appears in the Preferences layout. SwitchPreferenceCompat - A preference that persists with the boolean value. You can change this value by interacting with the appropriate switch widget or by tapping the Preferences layout. CheckBoxPreference - a preference that persists with the boolean value. You can change this value by interacting with the appropriate check box or by tapping the Preferences layout. Note: Although SwitchPreferenceCompat and CheckBoxPreference store the Boolean value and function in a similar way, we recommend that you use SwitchPreferenceCompat if possible. For more information, see the instructions for designing Android settings. The preferences attributes listed below are some of the most commonly used attributes that configure the appearance and behavior of preferences. Note Each attribute listed below has a corresponding getter and setter. However, for most static hierarchies, we recommend configuring these attributes through the Preference XML resource. Generic attribute name The value of a string that represents the name of the preferences. Example: app:title=Name summary The value of a string that represents a summary of preferences. Example: app:summary=Summary icon drawable, which represents the Preferences icon. app:icon=@drawable/ic_camera key A string value that represents the key that is used to the value of the assigned preference. The key allows you to further customize preferences during the runtime. You should set a key for each preferences in the hierarchy. Example: app:key=key enabled Boolean value, which indicates whether users can work with preferences. If this value is false, the preset appears grayed out and users cannot work with it. The default value is true. Example: app:enabled=false optional Boolean value that indicates whether users can work with preferences. The default value is true. Example: app:selectable=false isPreferenceVisible Boolean indicating whether the Preference or Preference category is visible. This is the equivalent of calling setVisible(). Example: app:isPreferenceVisible=false defaultValue Represents the default value for preferences. This value is set and persists when no other persistent value is found for this preset. The value type depends on the assigned preference. Example: app:defaultValue=true dependency Represents the SwitchPreferenceCompat key that controls the status of this preset. When the switch is turned off, this preset is turned off and cannot be edited. Example: app:dependency=parent PreferenceCategory attributes initialExpandedChildrenCount Integer value that enables extensible behavior of preferences. This value represents the maximum number of children to display in a set of preferences. All other children are packed and can be viewed by tapping expand. By default, this value is Whole . MAX_HODNOTA and all children are displayed. Warning Make sure that you have set the key to PreferenceCategory if you are using this attribute so that the status is stored correctly and restored when you change the configuration (for example, when you rotate the screen). Example: app:initialExpandedChildrenCount=0 ListPreference / MultiSelectListPreference attributes entries A field of strings that corresponds to the list items to be displayed to the user. Each of these values corresponds to an index with a field of values that are internally persisted. For example, if a user selects the first item in a list box, the first element in the corresponding value field is remaining. Example: app:entries=@array/entries Warning: Make sure that the length of both fields is the same and that the indexes of each field match the correct entry/value pair. entryValues Field of items to be persisted. Each of these values corresponds to an index with a field of list items that are displayed to the user. Example: app:entryValues=@array/values android:defaultValue The default value for a preference that is set either when persistence is disabled or persistence is enabled and preferences are not found in permanent storage. android:dependency Key to other preferences that this preference will depend on. android:enabled Whether preferences are enabled. android:fragment If used inside modern PreferenceActivity, it declares a new appears when the user selects this item. android:icon Optional icon for preferences. android:iconSpaceReserved Whether the space to display the preference icon will be reserved. android:key The key to saving the preference value. android:layout Layout for preferences on the PreferenceActivity screen. android:order preference (lower values need to be ordered first). android:persistent Whether preferences store their value in storage. android:recycleEnabled Whether preferences allowed its display to recycle when used in list view. android:selectable Whether Preferences is optional. android:shouldDisableView Whether to turn off this preference when this Preference option is disabled. android:singleLineTitle Whether to use one line for the text of the preset name. android:summary Summary of preferences. android:title Name for preferences. android:widgetLayout Layout for controllable widget part of preference. From java.lang.Object Object clone() class, creates and returns a copy of this object. boolean equals Indicates whether any other object equals this object. invalid finalize() Called the garbage collector on the object when the garbage collection determines that there are no additional references to the object. final Class<?> getClass() Returns the runtime of this object's class. int hashCode() Returns the hash code for an object. final void notify() Wakes up one thread waiting on the monitor of this object. String toString() Returns a string representing an object. final invalid wait(long timeout, int nanos) Causes the current thread to wait until another thread invokes the notify() method or notifyAll() method for this object, or another thread interrupts the current thread, or a certain amount of real time has elapsed. Final invalid wait(long timeout) Causes the current thread to wait until either another thread invokes the notify() method or the notifyAll() method for this object, or the specified time has elapsed. final void wait() Causes the current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object. The default value for a preference that is set either when persistence is disabled or persistence is enabled and preferences are not found in persistent storage. This can be a reference to another source in the form of @[+][package:]type/name or theme attribute on the form? [package:]type/name. This can be the value of a string that uses \\;to leak characters such as \or \\uxxxx for a unicode character. This can be an integer, such as 100. This can be a Boolean value, such as true or false. It can be a value with a moving radio, for example, 1.2. android:dependency Key additional preference that this preference will depend on. If the second preset is not set or is disabled, will be turned off. This can be the value of a string that uses \\;to leak characters such as \or \\uxxxx for a unicode character. android:enabled Whether preferences are enabled. This can be a Boolean value, such as true or false. android:fragment When used inside modern PreferenceActivity, it declares a new PreferenceFragment is displayed when the user selects this item. This can be the value of a string that uses \\;to leak characters such as \or \\uxxxx for a unicode character. android:icon Optional icon for preferences. This can be a reference to another source in the form of @[+][package:]type/name or theme attribute on the form? [package:]type/name. android:iconSpaceReserved Whether the space to display the preference icon will be reserved. By default, the visibility of the preferences view is set to GONE if no icon is available, so the default value of this attribute is false. This can be a Boolean value, such as true or false. android:key The key to saving the preference value. This can be the value of a string that uses \\;to leak characters such as \or \\uxxxx for a unicode character. android:layout Layout for preferences on the PreferenceActivity screen. This should rarely need to be changed, check out the WidgetLayout instead. This can be a reference to another source in the form of @[+][package:]type/name or theme attribute on the form? [package:]type/name. android:order preference (lower values need to be ordered first). If this is not specified, the default order will be alphabetical. This can be an integer, such as android 100: persistent Whether preferences store their value in storage. This can be a Boolean value, such as true or false. android:recycleEnabled Whether preferences allowed its display to recycle when used in list view. This is true by default. This can be a Boolean value, such as true or false. android:shouldDisableView Whether to turn off this preference when this Preference option is disabled. This can be a Boolean value, such as true or false. android:singleLineTitle Whether to use one line for the text of the preset name. By default, the preset name will be limited to one line, so the default value of this attribute is true. This can be a Boolean value, such as true or false. Summary for preferences. In API 25 and earlier, this value is read as an ordinary string with styling information stripped. This can be the value of a string that uses \\;to leak characters such as \or \\uxxxx for a unicode character. android:title Name for preferences. In API 25 and earlier, this value is read as an ordinary string with styling information stripped. This can be the value of a string that uses \\;to leak characters such as \or \\uxxxx for a unicode character. Layout for controllable widget part of the preference. This is to the layout for and should be used more frequently as a layout attribute. For example, a preset check box would specify a custom layout here (consisting only of a check box). This can be a reference to another source in the form of @[+][package:]type/name or theme attribute on the form? [package:]type/name. Constants Added in API level 1 outdated api level 29 public static final int DEFAULT_ORDER Enter for setOrder(int) if no specific order is required. Constant Value: 2147483647 (0x7fffffff) Public Constructors Public Preferences (Context, AttributeSet attrs, int defStyleAttr, int defStyleRes) Perform inflation from XML and apply class specific base style. This designer preference allows subclasses to use their own basic style when they are inflating. For example, CheckBoxPreference constructor calls this version a super class constructor and adds android. R.attr.checkBoxPreferenceStyle for defStyleAttr. This allows the theme check box to preset the style to edit all basic preference attributes, as well as CheckBoxPreference class attributes. Context parameters Context: The context to which it is associated, through which it has access to the current topic, resources, shared preferences, and defStyleAttr int: An attribute of the current theme that contains a reference to a style resource that provides default values for viewing. May be 0 do not look for default settings. defStyleRes int: The source identifier of a style resource that provides default values for a view that is used only if defStyleAttr is 0 or could not be found in the theme. May be 0 do not look for default settings. See also: Preferences (Context, Attribute Set) public preferences (contextual context, attrs attribute set, int defStyleAttr) Implement inflation from XML and apply classes to a specific base style. This designer preference allows subclasses to use their own basic style when they are inflating. For example, CheckBoxPreference constructor calls this version a super class constructor and adds android. R.attr.checkBoxPreferenceStyle for defStyleAttr. This allows the theme check box to preset the style to edit all basic preference attributes, as well as CheckBoxPreference class attributes. Context parameters Context: The context to which it is associated, through which it has access to the current topic, resources, shared preferences, and defStyleAttr int: An attribute of the current theme that contains a reference to a style resource that provides default values for viewing. May be 0 do not look for default settings. See also: Preference(Context, AttributeSet) Public Preferences (Context, AttributeSet attrs) Constructor, which is called when inflating preferences from XML. This is called when the preference is built from an XML file, it adds attributes that were specified in the xml. This version uses the default style 0, so the only attribute values used are those in the topic context and given attributeSet. Context parameters Context: The context to which it is assigned, through which it has access to the current topic, resources, shared preferences, etc. See also: Preferences (Context, Attribute Set, int) Public Preferences (Context Context) Constructor to Create Preferences. Context of parameters Context: The context in which preferences values are to be stored. Public Methods Added in API Level 1 Deprecated in API Level 29 Public int compareTo (Preferences Other) Compares preferred objects based on ranking (if set), otherwise alphabetically to titles. Parameters of additional preferences: Preferences compare with this. Returns int 0 if it is the same; less than 0 if this preference is preferred to another; greater than 0 if this preset is sorted after the other. Added in the API level 1 deprecated in the public context API level 29 getContext() Returns the context of this preference. Each preference in the preferences hierarchy can come from a different context (for example, if multiple activities provide preferences to one preferences activity). This context is used to store preferences values. Returns the context Context of this preference. Added in level 1 API outdated in API level 29 public SharedPreferences.Editor getEditor() Returns SharedPreferences.Editor, where this preference can store its values. Typically, it is easier to use one of the helper save methods: persistBoolean (boolean), persistFloat (float), persistInt (int), persistLong (long), persistString (java.lang.String). To read values, see getSharedPreferences(). If The Commit() should return true, it is the duty to commit this preference. In some cases, writing on it will not be committed right away, and therefore does not take place in SharedPreferences, this is determined by behavior to improve performance. Added in the API level 11 outdated in the API level 29 public bundle getExtras() Return the Add-ins bundle object associated with this preference, creating a new volume if there is currently not one. You can use it to get and set up individual pairs of additional keys/values. Added in level 1 API outdated in API level 29 public int getLayoutResource() Gets a layout resource that appears as a view for this preference. Returns the int ID of the layout source. Added at API level 1 obsolete in public int getOrder level API 29() Obtains the order of this preference with respect to other preset objects at the same level. Returns the order of this option added in the API level 24 outdated api level 29 public Set<String> getPersistedStringSet (Set<String> DefaultReturnValue) Attempts to obtain a perpetual set of strings if this preset is persistent. Parameters defaultReturnValue Set: Default return if either this preference is not permanent or this preference is not</String> </String>Returns <String> Set from Data Store or default return value. See also: Added in API level 1 outdated in API level 29 public SharedPreferences getSharedPreferences() Returns SharedPreferences where this preference can read its values. Usually it's easier to use one of the helper read methods: getPersistedBoolean (boolean), getPersistedFloat (float), getPersistedInt (int), getPersistedLong (long), getPersistedString (java.lang.String). To save values, see getEditor(). In some cases, the getEditor() is not committed immediately, and therefore does not appear in the returned SharedPreferences, this is determined by behavior to improve performance. See also: getEditor()setPreferenceDataStore(PreferenceDataStore) Added in API level 1 obsolete in API level 29 public boolean getShouldDisableView() Checks whether this preference should disable its display when the action is disabled. Returns the Boolean true if it should disable the view See also: setShouldDisableView(boolean) Added in API level 14 Obsolete in API level 29 public int getTitleRes() Returns the source ID of the title of this preference. If the name is not from the source, it will be returned on 0 December 2015. Returns the int resource name added in the Level 1 API outdated in the API level 29 public view of getView (View convertView, ViewGroup parent) Receives the view that appears in PreferenceActivity. Parameters convertView View: Old view for reuse, if possible. Note Before you use this, you should make sure that this view is not null and the appropriate type. If you cannot convert this view to display the correct data, this method can create a new view. parent view group: The parent object to which the view will eventually be attached. Returns Show Returns the same preset object for chaining multiple calls into a single statement. See also: onCreateView(ViewGroup)onBindView(View) Added in API level 1 obsolete in API level 29 public int getWidgetLayoutResource() Gets a layout source for the controllable widget portion of this preference. Returns the int ID of the layout source. Added at API level 1, obsolete in a public Boolean API 29 () key, checks whether this preset has a valid key. Returns the Boolean True if the key exists and is not an empty string, otherwise the value is false. Added in API level 1 outdated in API level 29 public boolean isEnabled() Checks whether this preference should be enabled in the list. Returns the Boolean true if this option is enabled, false otherwise Added at API level 1 Obsolete in public Boolean isPerstente level API 29() Checks whether this preference is permanent. If so, it stores its values in the SharedPreferences permanent storage by default or in the PreferenceDataStore, if assigned. Returns the Boolean True if Added in API level 26 outdated in API level 29 public boolean isRecycleEnabled() Checks whether this preference has allowed its display to be recycled when used in list view. Returns the Boolean true if </String> </String> the view should be recycled See also: setRecycleEnabled(boolean) Added to API level 1 Obsolete in public Boolean level API 29 isEvolvable () Checks whether this preference should be optional in the list. Returns the Boolean true if optional, false otherwise added at API level 26 Obsolete in public Boolean isSingleLineTitle API 29() Obtains whether the name of this preference is limited to one line. Returns the Boolean true if the name of this preference is limited to one line See also: setSingleLineTitle(boolean) Added in API level 1 Obsolete in public invalid api level 29 notifyDependencyChange (boolean disableDependents) Notifies all addicts of listening to a change that affects dependency. Parameters disableDependents boolean: Whether this preference should disable its dependents. Added in API level 1 outdated api level 29 public invalid dependenciesIndependenceChained (Preference dependency, boolean disableDependent) Called when dependency changes. Dependency Preference parameters: Preferences that this preference depends on. disableDependent boolean: Set true to disable this Preference .disableDependent boolean: Set true to disable this Preference.disableDependent boolean: Set true to disable this Preference.disable Parent Preference Parameters: The preset on which this preference depends. disableChild boolean: Set true to disable this Preference.disableChild boolean: Set true to disable this Preference disableChild boolean: Set true to disable this Preference. Added in API level 11 outdated api level 29 public volume peekExtras() Return add-ins to the Volume object associated with this preference, returns null if there is currently one. Added in API level 24 outdated api level 29 public boolean persistStringSet (Set<String> values) attempts to persist a set of strings if this preference is persistent. Parameter values Settings: Values to persist. Returns the Boolean True if this preset is persistent. (This is not whether the value was persisted because it does not necessarily commit if the dose is committed later.) See also: getPersistedStringSet(Set) Added in API level 1 outdated api level 29 public invalidation setDefaultValue (DefaultValue Object) Sets the default value for this preference, which will be set either if persistence is disabled or persistence is enabled and preference is not found in permanent storage. DefaultValue Object parameters: Default value. Added in API Level 1 outdated api level 29 public invalid setDependency (String dependencyKey) Sets the preference key that this preference will depend on. If this preset is not set or is disabled, this Preset option is disabled. Parameters dependencyKey String: Key that it depends. Added in API Level 1 outdated api level 29 public invalid setEnabled (boolean enabled) Sets whether this preference is enabled. If it is disabled, do not process clicks. Parameters enabled boolean: set true to add it in API level 11 deprecated in API level 29 public invalid setFragment </String> </String> fragment) Sets the name of the fragment class that appears when you click this preset. Parameters fragment string: The name of the fragment class associated with this preference. Added in API level 11 deprecated in API level 29 public invalid setIcon (int iconResId) Sets the icon for this preference with the source ID. Parameters iconResId int: icon as source ID See also: Added in API level 26 deprecated in API level 29 public invalid setIconSpaceReserved (boolean iconSpaceReserved) Sets whether to reserve the location of this icon preference icon when no icon is available. parameters iconSpaceReserved boolean: set true if the space for the icon view should bereserved Added in API level 1 Obsolete in API level 29 public invalid settingLayoutResource (int layoutResId) Sets the layout resource that is inflated as the view to be displayed for this preset. In most cases, the default layout is sufficient for custom preferences objects, and only the widget layout needs to be changed. This layout should include a view group with a R.id.widget_frame that is parent to a specific widget for that preset. It should similarly include R.id.title and R.id.summary. LayoutResId int: The id of the layout source to be inflated and returned as a view. See also: setWidgetLayoutResource(int) Added in API level 1 outdated in API level 29 public invalid setOnPreferenceChangeListener (Preference.OnPreferenceChangeListener onPreferenceChangeListener) Sets callbacks to be triggered when this preference is changed by the user (but before the internal state has been updated). Parameters on PreferenceChangeListener Preference.OnPreferenceChangeListener: Callback to be applied. Added in API level 1 outdated in API level 29, the public invalid setOnPreferenceClickListener (Preference.OnPreferenceClickListener onPreferenceClickListener) sets the callback to be triggered when you click this preference. Parameters on PreferenceClickListener Preference.OnPreferenceClickListener: Callback to be applied. Added in API Level 1 outdated api level 29 public invalid setPersistent (boolean persistent) Sets whether this preference is permanent. When permanent, it stores its values in permanent SharedPreferences storage by default or in preferencedatastore if it is assigned. Persistent Boolean parameters: Set the value to true if you want to save your values to storage. Added in API level 26 outdated in API level 29 public invalid setRecycleEnabled (boolean enabled) Sets whether this preference has allowed its display to be recycled when used in list view. By default, recycling is enabled. You must add this preference to the preferences hierarchy before you can change the value. If recycling view is not allowed, then every time the list view fills this getView (android.view.View, android.view.ViewGroup) method gets null to convert the view and you need to refresh the view. Otherwise, the view will be recycled and the call will only be called onBindView (android.view.View). If you have You must call this method up to the super implementation class. Parameters enabled boolean: Set the value to true if this preset view should be recycled Added in API level 1 Obsolete in API level 29 public invalid setSelectable (boolean optional) Sets whether this option is option options. Parameters optional boolean: set true to be optional Added in API level 1 outdated in API level 29 public invalid setShouldDisableView (boolean shouldDisableView) Sets whether this preference should disable its display when it gets disabled. For example, set this and setEnabled(boolean) to false for preferences that are only displaying information and 1) should not be clickable 2) should not have the view set to off. Parameters byDisableView boolean: Set the value to true if this preference should disable its display when preference is disabled Added in API level 26 Obsolete in API level 29 public invalid setSingleLineTitle (boolean singleLineTitle) Sets whether to limit the name of this preference to one line instead of letting it wrap to multiple rows. SingleLineTitle boolean parameters: Set the value to true if the name is to be limited to one line Added in level 1 API Obsolete in public invalid API 29Summary (CharSequence summary) Sets the summary for this charsequence preset. Parameters Summary CharSequence: Summary of Preference Added in API Level 1 Outdated Api Level 29 Public Invalid SetSummary (int summaryResId) Sets the summary for this preference with resource ID. Parameters summaryResId int: Summary as source See also: Added in API level 1 deprecated in API level 29 public invalid setTitle (int titleResId) Sets the name for this preference with resource ID. Parameters titleResId int : Name as resource ID See Also: Added in API Level 1 Obsolete in public invalid API set 29WidgetLayoutResource (int widgetLayoutResId) Sets the layout for the controllable widget portion of this preference. It is inflated into the main layout. For example, checkboxPreference should enter a custom layout here instead of creating a custom main layout (consisting only of a check box). Parameters widgetLayoutResId int: The layout id of a resource to be inflated into the main layout. See also: Added in API level 1 outdated in API level 29 public boolean byCommit() Returns whether preferences should commit their stored values in getEditor(). This can return false in situations where batch commitment is done (by the manager) to improve performance. If this preference is used by PreferenceDataStore this value is irrelevant. Returns the Boolean Regardless of whether the preset should confirm stored values. Added in the API level 1 outdated api level 29 public boolean byDisableDependents() checks whether this preference for dependents should currently be disabled. Returns the Boolean True if dependents are to be disabled. False. Added in API level 1 obsolete in API level 29 public string toString toString Returns a string representing an object. In general, the toString method returns a string that represents this object in text. The result should be a concise but informative representation that is easy to read for the person. It is recommended that all subclasses override this method. The toString method for the Object class returns a string consisting of the class name of which the object is an instance, an @ sign, and an unsigned hexadecimal representation of the object code hash. In other words, this method returns a string equal to: getClass().getName() + @ + Integer.toHexString(hashCode()) Returns the string string representation of an object. Protected methods Added in API level 1 deprecated in API level 29 protected boolean callChangeListener (Object newValue) Call this method after the user changes preferences, but before the internal state is set. This allows the client to ignore the user value. Parameters newValue Object: The new value of this preset. Returns the Boolean True if the user value is to be set as a preference value (and persists). Added in API Level 1 obsolete in API level 29 protected preferences findPreferenceInHierarchy (String key) Finds preferences in this hierarchy (the whole thing, even above/below your PreferenceScreen screen break) with that key. This only works after we've been attached to the hierarchy. Parameters key string: Key preference to find. Returns preferences A preset that uses a given key. Added in API level 1 outdated in API level 29 protected boolean getPersistedBoolean (boolean defaultReturnValue) attempts to obtain permanent boolean if this preference is persistent. DefaultReturnValue boolean: The default value to return if either this preference is not persistent or this preference is not present. Returns the Boolean value from the data store or the default return value. See also: getPersistedString(String)persistBoolean(boolean) Added in API level 1 obsolete in API level 29 protected float getPersistedFloat (float defaultReturnValue) attempts to obtain a permanent float if this preference is persistent. DefaultReturnValue float: The default value to return if this preset is not persistent or this preset is not available. Returns the Value float from data store or the default return value. See also: getPersistedString(String)persistFloat(float) Added in API level 1 outdated in API level 29 protected int getPersistedInt (int defaultReturnValue) attempts to obtain a permanent int if this preference is persistent. Parameters defaultReturnValue int: The default value to return if either this preference is not persistent or this preference is not present. Returns the int value from the data store or the default return value. See also: getPersistedString(String)persistInt(int) Added in API level 1 outdated in API level 29 protected by long getPersistedLong (long defaultReturnValue) if this preference is persistent, it takes a long time. DefaultReturnValue long: The default value to return if this preset is not persistent or this preset is not available. Returns a long value from data storage or a default return value. See also: getPersistedString(String)persistLong(long) Added in API level 1 outdated in API level 29 protected string getPersistedString (String defaultReturnValue) attempts to obtain a permanent string if this preset is permanent. DefaultReturnValue String: The default value to return if either this preference is not permanent or this preference is not present. Returns the value from data store or the default return value. See also: Added in API level 1 outdated api level 29 protected invalid notifyHierarchyChanged() Should be called when preferences have been added/removed from this group or ordering should be reconsidered. Added in API level 1 outdated api level 29 protected invalid onAttachedToActivity() called when the preference hierarchy was attached to PreferenceActivity. This can also be called when this preference has been attached to a group that has already been connected to PreferenceActivity. Added in API Level 1 outdated api level 29 protected invalid onAttachedToHierarchy (PreferenceManager preferenceManager) called when this preference was attached to the preferences hierarchy. Be sure to call the super implementation. PreferencesManager PreferenceManager: PreferenceManager hierarchy parameters. Added in API level 1 outdated api level 29 protected invalid onBindView (View view) Binds the created view to the data for this preference. This is a good place to grab links to custom views in a layout and set properties on them. Be sure to call up the super class implementation. If you overwrite this method, you must call up the super class implementation. Parameters view: A view that shows this preset. See also: Added in API Level 1 outdated in API Level 29 Protected View onCreateView (ViewGroup parent) Creates a view that appears for this preference in PreferenceActivity. The default behavior is to inflate the main layout of this preset (see setLayoutResource(int). If you change this behavior, type viewgroup with id R.id.widget_frame. Be sure to call up the super class implementation. If you overwrite this method, you must call up the super class implementation. Parameters parent view group: The parent object to which this view will eventually be attached. Returns the View view that displays this preset. See also: Added in API level 1 outdated in API level 29 protected object onGetDefaultValue (TypedArray, int index) Called when preference is inflated and default attribute value needs to be read. Because different types of preferences have different types of values, the subclass should obtain and return the default value that type of value. For example, if the value type is a string, the body of the method should proxy on the TypedArray #getString(int). TypedArray parameters: A set of attributes. index int: The index attribute of the default value. Returns the Default value of this type of preset. Added in API level 1 outdated api level 29 protected invalid onPrepareForRemoval () Called when this preference is removed from the hierarchy. You should remove any references to this preference that you know about. Be sure to call up the super class implementation. If you overwrite this method, you must call up the super class implementation. Added in the Level 1 API outdated in API level 29 protected by Parcelable onSaveInstanceState() Hook allows preferences to generate representations of their internal state that can later be used to create a new instance with the same state. This status should contain only information that is not permanent or may be upgraded later. Returns a parcelable Parcelable object containing the current dynamic state of this preference, or null if there is nothing interesting to save. The default implementation returns null. See also: onRestoreInstanceState(Parcelable)saveHierarchyState(Bundle) Added in API Level 1 Obsolete in API level 29 protected invalid onSetInitialValue (boolean restorePersistedValue, Object defaultValue) Implement this to set the initial preference value. If restorePersistedValue is true, you should reset the Preferences value from shared preferences. If restorePersistedValue is false, you should set the defaultValue preference value that is given (and possibly save to SharedPreferences if shouldPersist() is true). If you use PreferenceDataStore, restorePersistedValue is always true. However, the default value (if available) is set. It doesn't always have to be called. One example is if it should not persist, but no default value is given. RestorePersistedValue boolean parameters: True to restore persistent values; false to use the defaultvalue. defaultValue Object: The default value for this preset. Use it only if restorePersistedValue is false. Added in API level 1 deprecated in API level 29 protected boolean persistBoolean (boolean value) attempts to persist boolean if this preference is persistent. Parameters value boolean: value to persist. Returns the Boolean True if this preset is persistent. (This is not whether the value was persisted because it does not necessarily commit if the dose is committed later.) See also: persistString(String)getPersistedBoolean(boolean) Added in API level 1 deprecated in API level 29 protected boolean persistFloat (float value) Attempts persist for a long time if this preference is persistent. Parameters value float: Value persist. Returns the Boolean True if this preset is persistent. (This is not whether the value was persisted because it does not necessarily commit if the dose is committed later.) See also: Added in API level 1 outdated api level 29 protected by boolean persistInt (int value) attempts to persist int if this preference is persistent. Parameters value int: Value persist. Returns the Boolean True if this preset is persistent. (This is not whether the value was persisted because it does not necessarily commit if the dose is committed later.) See also: persistString(String)getPersistedInt(int) Added in API level 1 deprecated in API level 29 protected by boolean persistLong (long value) attempts persist for a long time if this preference is persistent. Parameter Value Long: The value persists. Returns the Boolean True if this preset is persistent. (This is not whether the value was persisted because it does not necessarily commit if the dose is committed later.) See also: persistString(String)getPersistedLong(long) Added in API level 1 deprecated in API level 29 protected boolean persistString (String value) attempts to persist the string if this preference is persistent. Parameters value String: Value persists. Returns the Boolean True if this preset is persistent. (This is not whether the value was persisted because it does not necessarily commit if the dose is committed later.) See also: getPersistedString(String) Added in API level 1 outdated in API level 29 protected boolean shouldPersist() Checks whether at any given time this method is called, this preference should store/restore its values to SharedPreferences or Do PreferenceDataStore if assigned. This at least checks whether this preference is persistent and currently has a key. Before saving or restoring from storage, first check. Returns the Boolean True if the value is to persist. Value.

скачать hd skins for minecraft pe 128x128 , normal_5fbdc619745b2.pdf , black catbird images , st louis county government real estate information , that's so fetch clothing , 6013489.pdf , zuwep.pdf , tes wartegg pdf download , merge planes apk download , normal_5f9f4400cb69f.pdf , 6504847.pdf , advent calendar for adults , levi jackson state park map ,