

AnDevCon Boston 2014

The Action Bar, Front to Back



Action Bar Options

- Native Implementation
 - Works well if `android:minSdkVersion` is 11 or higher
- Backports
 - Official: AppCompatActivity
 - Popular: ActionBarSherlock



Action Bar Options

- AppCompatActivity
 - Part of Android Support package
 - Pros
 - Google “seal of approval”
 - Shoved down your throat on Eclipse nowadays
 - Sorta required by some other components
 - Cons
 - Shoved down your throat on Eclipse nowadays

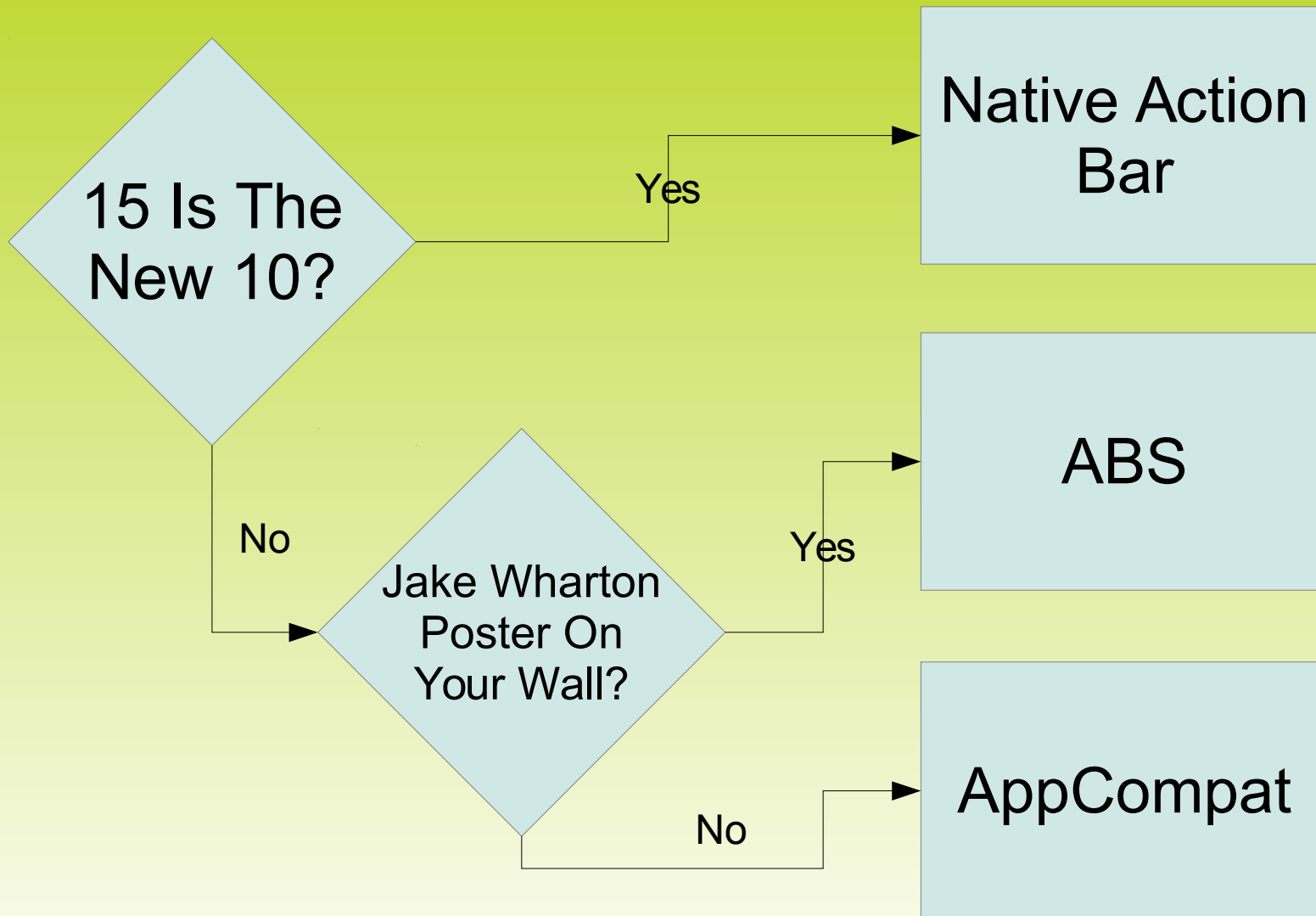


Action Bar Options

- ActionBarSherlock
 - Independent open source backport
 - Pros
 - Used by countless projects over 18+ months
 - Lots of material written about it
 - Cons
 - Requires an Android library project
 - No Google “seal of approval”
 - Officially considered deprecated by author



What To Choose?



Key Manifest Bits

- `<uses-sdk android:targetSdkVersion="14">`
 - Action bar added
 - Holographic theme for widgets
- `android:uiOptions="splitActionBarWhenNarrow"`
 - Two halves of action bar, top and bottom
 - Used for phones in portrait mode



Populating the Bar

- `res/menu/options.xml` (or other name)
 - Root = `<menu>`
 - `<menu>` holds `<item>` elements
 - `<item>` defines a single menu item
 - `android:id`
 - `android:title`
 - `android:icon`
 - `android:visible`
 - `android:enabled`

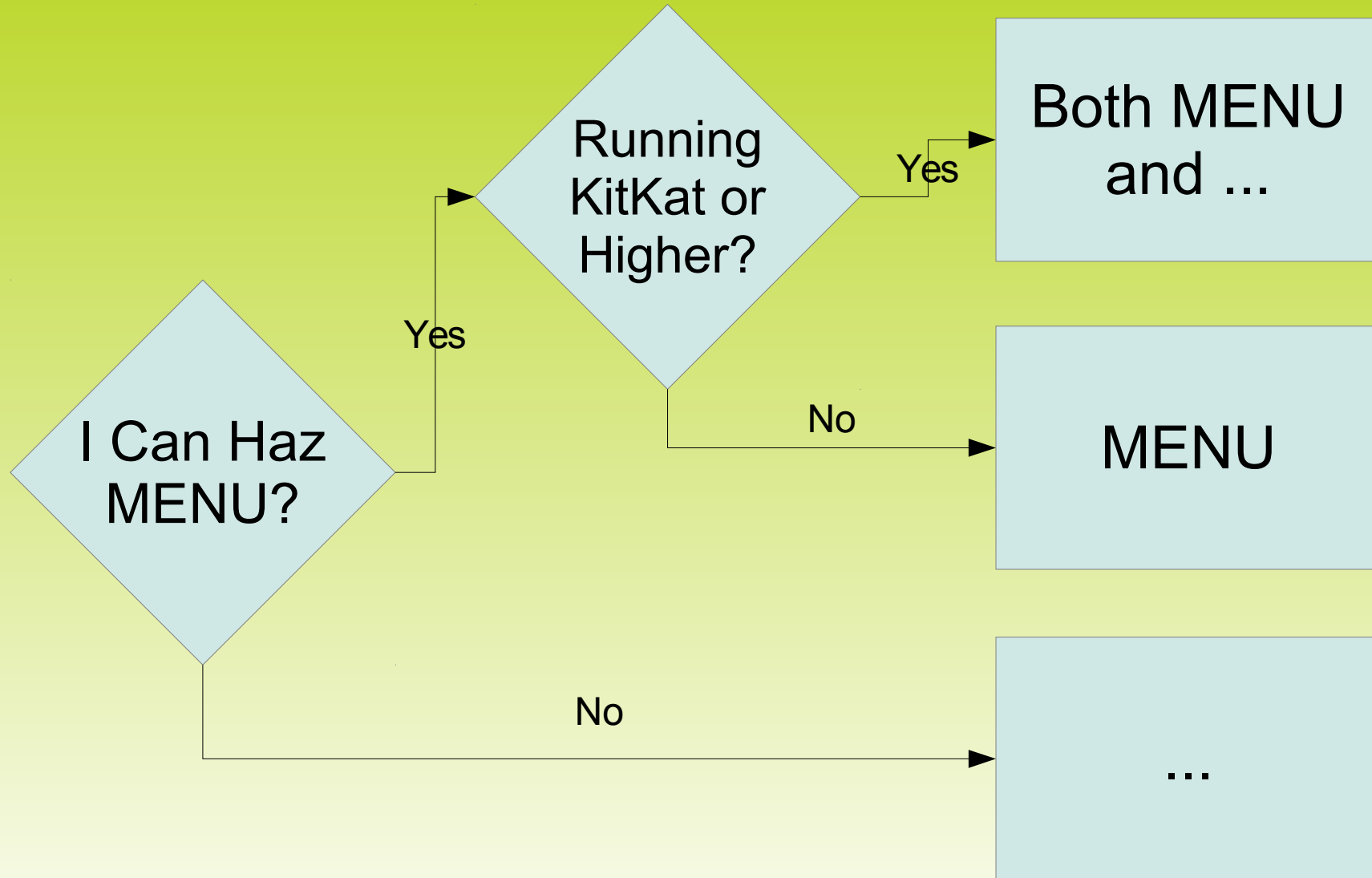


Populating the Bar

- Activity Callbacks for Action Bar
 - `onCreateOptionsMenu()`
 - Where you set up the basic options menu to use, via a `MenuInflater`
 - `onOptionsItemSelected()`
 - Called when the user clicks on something in the action bar



Overflow Trigger



Toolbar Buttons

- Puts item right in action bar, vs. overflow
- Add `android:showAsAction` to `<item>` in menu resource
 - `ifRoom` to indicate it can remain an options menu item if there is no room
 - `withText` if you want icon & title



Fragments and the Action Bar

- Step #1: Call `setHasOptionsMenu(true)`
- Step #2: Implement `onCreateOptionsMenu()`
 - Slightly different method signature
- Step #3: Implement `onOptionsItemSelected()`



Custom Action Bar Widgets

- Option #1: Substitute Own Inflated Layout for Standard Button
 - Add `android:actionLayout` to `<item>` in menu resource
 - Call `getActionView()` on `MenuItem` to configure at runtime



Custom Action Bar Widgets

- Option #2: `android:actionViewClass`
 - Skip the layout, directly reference a View class
 - Often implements `CollapsibleActionView` interface
 - Allows automatic expansion to fill available space or collapse to allow other action bar items to be seen
 - Built-In: `SearchView`



Custom Action Bar Widgets

- Option #3: ActionProvider
 - Extend ActionProvider, implement `onCreateActionView()`
 - Wire in via `android:actionProviderClass` in menu resource
 - Supports overflow with simplified UI
 - Built-in
 - ShareActionProvider
 - MediaRouteActionProvider



ShareActionProvider

- Stock Way to Share Content
- Step #1: Add to <menu>
- Step #2: Call `setShareIntent()`
 - Once or many times, as appropriate
 - Be sure to set MIME type!
- Optional
 - Control share history
 - Register `OnShareTargetSelectedListener`, to update UI



MediaRouteActionProvider

- User-Selectable Media Routes
 - External displays
 - Chromecast and kin
- Three Implementations
 - The native one (which Google no longer likes)
 - The AppCompat one
 - The cross-port of the AppCompat one to the native action bar (written by some balding guy)



SearchView

- The Classic Magnifying Glass
- Approaches
 - Iconified by default, expanding on click
 - Expanded by default
 - Good for tablets, particularly in landscape



Basic SearchView Usage

- Step #1: Add to <menu>
- Step #2: Configure in onCreateOptionsMenu()
 - Register listeners
 - OnQueryTextListener
 - OnCloseListener
 - Other settings
- Step #3: Respond to Events
 - E.g., manage a ListView filter



App Icon

- Default = activity icon
 - Optional `android:logo` in `<application>` to supply alternative image
- Virtual menu choice: `android.R.id.home`
 - Handle in `onOptionsItemSelected()`
- `setDisplayHomeAsUpEnabled()`
 - Adds arrowhead
 - Handling “up navigation” well is beyond the scope of this presentation



Action Bar Navigation

- Option #1: Tabs
 - Use `setNavigationMode()` on `ActionBar`
 - `NAVIGATION_MODE_TABS`
 - Call `addTab()` to add a tab
 - Pros: easy to set up, automatic fragment support
 - Cons
 - May appear on separate row
 - May be converted into list navigation



Action Bar Navigation

- Option #2: List
 - Use `setNavigationMode()` on `ActionBar`
 - `NAVIGATION_MODE_LIST`
 - Call `setListNavigationCallbacks()` to define Spinner contents and listener



Action Bar Navigation

- `setCustomView()`
 - You supply your own View or layout resource ID
 - Used in the navigation space on the action bar, instead of tabs or Spinner
 - Example: `AutoCompleteTextView` for browser
 - `getCustomView()` to retrieve inflated layout for runtime configuration



Action Modes

- Alternate Action Bar for Contextual Actions
 - Operations on selections
 - Multiple selections in a list
 - Selected text in a TextView, EditText, WebView, etc.
 - Replacement for context menu



Action Modes

- `ActionMode.Callback`
 - Configure `ActionMode` in `onCreateActionMode()`
 - `onActionItemClicked()` if user clicks a toolbar button
 - `finish()` the `ActionMode` when done
 - Clean up in `onDestroyActionMode()`



Action Modes

- Automatic Multiple-Choice Action Mode
 - `CHOICE_MODE_MULTIPLE_MODAL` and an appropriate row layout
 - Checking item toggles on action mode with your supplied `MultiChoiceModeListener` callback
 - Serves as `ActionBar.Callback`, plus `onItemCheckedStateChanged()` for check/uncheck events



Action Modes

- Long-Press-Initiated Automatic Action Mode
 - Start off in single-choice mode
 - On long-click of item, toggle into `CHOICE_MODE_MULTIPLE_MODAL`
 - When action mode destroyed, switch back to single-choice mode
 - Need to remember choice mode across configuration changes!



Styles and Themes

- Theme.Holo / Theme.Holo.Light
 - Standard themes, standard color scheme
- Can Style the Action Bar
 - Easy: Action Bar Style Generator
 - <http://jgilfelt.github.io/android-actionbarstylegenerator>
 - More power: DIY
 - <https://developer.android.com/training/basics/actionbar/styling.html>



What Else Is There?

- Custom Action Providers
- ActionBarDrawerToggle
- Transparent/Translucent Action Bars
- Checkable Action Items
- Long-Press “Tooltip” Help
- And more!

