

Much Manifest Merging



“Mommy, Where Do Manifests Come From?”

- Gradle build files
- Sourcesets
 - Build types and product flavors
 - `androidTest` (if applicable)
 - `main`
- AARs and library modules
 - Priority based on order in dependencies



“Um, Mommy, That Seems Complicated...”

- Scenario #1: Debug-Only Activity
 - Diagnostic activity, to view object caches sans debugger
 - Java code, resources, and manifest entry in debug sourceset
 - Manifest entry: mostly an `<activity>` element with `<intent-filter>`
 - Net: diagnostic activity only in debug builds



“Um, Mommy, That Seems Complicated...”

- Scenario #2: Flavor-Specific Maps Metadata
 - Google: Play Services <meta-data> elements
 - Amazon: Maps ApiPolicy <meta-data> elements
 - Generic using OSMDroid: none of the above



“Um, Mommy, That Seems Complicated...”

- Scenario #3: Library Contributions
 - Requirements
 - Minimum SDK version
 - Required permissions, features
 - Components
 - Activities
 - Services
 - Broadcast Receivers
 - ContentProviders (maybe)



Outputs of Manifest Merging

- The Merged Manifest
 - (duh!)
 - `build/intermediates/manifests` in your app module(s)
 - Subdirectories by build variant (flavor, then build type)



Outputs of Manifest Merging

- Manifest Merger Report
 - `build/outputs/apk` of your app module(s)
 - One per build variant
 - Good news: human-readable text files
 - Bad news: somewhat inscrutable



```
application
ADDED from AndroidManifest.xml:9:5
MERGED from AndroidManifest.xml:9:5
MERGED from Merger:lib:unspecified:11:5
  android:label
    ADDED from AndroidManifest.xml:10:9
  android:allowBackup
    ADDED from AndroidManifest.xml:9:18
  android:uiOptions
    ADDED from AndroidManifest.xml:13:9
    ADDED from AndroidManifest.xml:13:9
  android:icon
    ADDED from AndroidManifest.xml:11:9
  android:theme
    ADDED from AndroidManifest.xml:12:9
receiver#com.commonware.android.merger.GcmBroadcastReceiver
ADDED from AndroidManifest.xml:14:7
  android:permission
    ADDED from AndroidManifest.xml:16:9
  android:name
    ADDED from AndroidManifest.xml:15:9
intent-filter#{applicationId}+com.google.android.c2dm.intent.RECEIVE
ADDED from AndroidManifest.xml:17:9
action#com.google.android.c2dm.intent.RECEIVE
ADDED from AndroidManifest.xml:18:11
  android:name
    ADDED from AndroidManifest.xml:18:19
category#{applicationId}
ADDED from AndroidManifest.xml:19:11
  android:name
    ADDED from AndroidManifest.xml:19:21
    INJECTED from AndroidManifest.xml:0:0
```



Element Matching

- Elements With `android:name`
 - Match if element name matches (e.g., “activity”) and `android:name` matches
- Elements Sans `android:name`
 - Match if element name matches (e.g., “supports-screens”)



General Merger Rules

- Rule #1: Gradle Über Alles
 - Example: one manifest has `targetSdkVersion`
 - Results
 - If overridden in `build.gradle`, Gradle wins
 - If not, the one manifest definition wins



General Merger Rules

- Rule #2: Disparate Elements Just Merge
 - Example
 - main has INTERNET permission
 - Library has ACCESS_FINE_LOCATION permission
 - Results
 - Merged manifest has both, as they are disparate
 - Match on element name
 - Do not match on android:name



General Merger Rules

- Rule #3: Disparate Attributes of Matching Elements Merged
 - Example
 - Library has `<activity>` element for `FooActivity`, with no `android:enabled` attribute
 - main has `<activity>` element for `FooActivity`, with `android:enabled="false"`
 - Results
 - Merged manifest has attributes for `<activity>` from both manifests



General Merger Rules

- Rule #4: Matching Attributes Merge
 - Example
 - Main has `WRITE_EXTERNAL_STORAGE` permission with `android:maxSdkVersion="19"`
 - Library has `WRITE_EXTERNAL_STORAGE` permission with `android:maxSdkVersion="19"`
 - Results
 - Merged manifest has `WRITE_EXTERNAL_STORAGE` permission with `android:maxSdkVersion="19"`



General Merger Rules

- Rule #5: Conflicting Attributes Break Build
 - Example:
 - main has `android:enabled="false"` for a library-supplied `<activity>`
 - Library explicitly has `android:enabled="true"` for that `<activity>`
 - Results
 - BOOM!



Specialty Rules

- `<uses-feature>`
 - If any manifest says the feature is required, it is required
 - If manifests only say the feature is not required, the element is included with `android:required="false"`
- `<uses-library>`: same rule



Controlling the BOOM!

- `tools:replace`
 - Put in higher-priority manifest
 - Build types/product flavors, `androidTest`, `main`
 - Comma-delimited list of other attributes in this element that conflict
 - Indicates that this element's values should be used to resolve the conflict



Controlling the BOOM!

- `tools:remove`
 - Put in higher-priority manifest
 - Build types/product flavors, `androidTest`, `main`
 - Comma-delimited list of attributes found in some lower priority manifest's edition of this element
 - Indicates that the attributes should be removed



Controlling the BOOM!

- `tools:node`
 - Put in higher-priority manifest
 - Build types/product flavors, `androidTest`, `main`
 - Controls merger of entire element, with respect to lower-priority manifest's edition of the element



Controlling the BOOM!

- `tools:node` Values
 - `merge` (default)
 - `replace` (use this element, not lower priority one)
 - `merge-only-attributes` (use this element's attributes, but merge child elements)
 - `remove` (have neither this element nor the lower-priority edition)
 - `removeAll` (wipe out all elements for this element name from lower-priority manifests)



General Conflict Guidance

- Libraries: Assume Nothing
 - Anything you put into the manifest can get overridden at app level
 - And `<uses-sdk>` seems to be ignored outright
 - Validate all your manifest assumptions (e.g., PackageManager) and fail fast
 - Does your BroadcastReceiver exist?
 - Are you running on a lower version of Android than expected?
 - Did you get a requested permission?



General Conflict Guidance

- Libraries: Keep Your Manifest Svelte
 - Only have elements, attributes that you really want
 - Do not explicitly set values that are already the default (e.g., `android:enabled="true"`)
 - Exception: `targetSdkVersion` (always have this)



General Conflict Guidance

- Apps: Override Libraries Carefully
 - Not every developer is attending this webinar and will follow the preceding guidance
 - Test thoroughly



Placeholders

- “Macros” for Manifests
 - `${}` syntax to refer to a placeholder
 - E.g., `{applicationId}`
 - Actual placeholder value determined at build time
 - E.g., based on Gradle build settings, build variant



Placeholders

- Example: GCM and `${applicationId}`
 - `<permission>`
 - `<uses-permission>`
 - `<category>`




```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.commonware.android.merger"
    android:versionName="1.0 Main">

    <permission android:name="${applicationId}.C2D_MESSAGE"
        android:protectionLevel="signature" />
    <uses-permission android:name="${applicationId}.C2D_MESSAGE" />

    <application android:allowBackup="true"
        android:label="@string/app_name"
        android:icon="@drawable/ic_launcher"
        android:theme="@style/AppTheme"
        android:uiOptions="splitActionBarWhenNarrow">
        <receiver
            android:name=".GcmBroadcastReceiver"
            android:permission="com.google.android.c2dm.permission.SEND" >
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.RECEIVE" />
                <category android:name="${applicationId}" />
            </intent-filter>
        </receiver>
    </application>

</manifest>
```



Placeholders

- Custom
 - `manifestPlaceholders` map
 - `defaultConfig`
 - Product flavors

