



RAD Studio™ 10 Seattle

Android Services
Jim McKeeth
Lead Developer Evangelist
Embarcadero Technologies

About Jim McKeeth

- Lead World Wide Developer Evangelist
- Host of Podcast at Delphi.org
- Longtime developer
 - Object Pascal, Java, C#, JavaScript, Objective-C, etc.
- Invented and patented swipe to unlock in 2000
 - US Patent # 8352745 & 6766456, etc.
- Improvisational performer with ComedySportz Boise



Why Android Services?

- Provide background functionality
- Can run even when app is closed
- Can start on boot or other events
- Can be accessible by multiple apps

Android Services vs. Threads

Android Services

- Cannot access the UI
- Runs in main thread of host app process
- Can contain multiple threads
- Used for long running processes

Threads

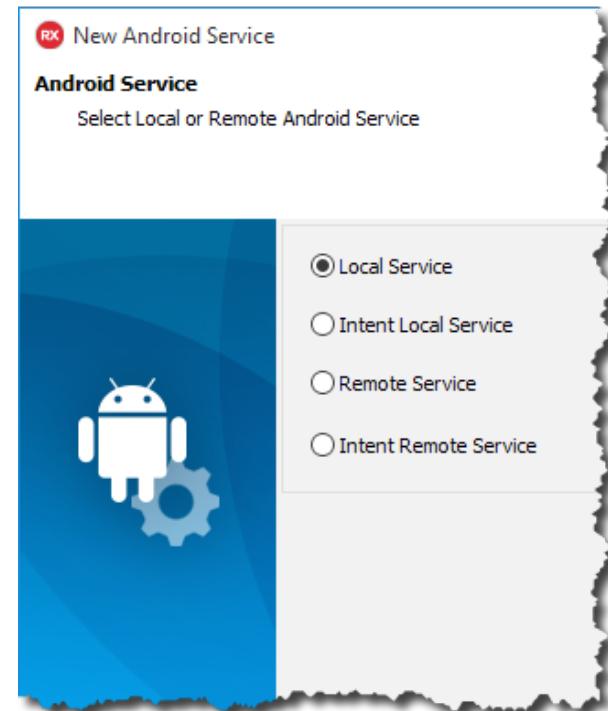
- Cannot access the UI
- Not accessible to other apps
- Terminated with app
- Used to improve responsiveness

Android Service Type Variations

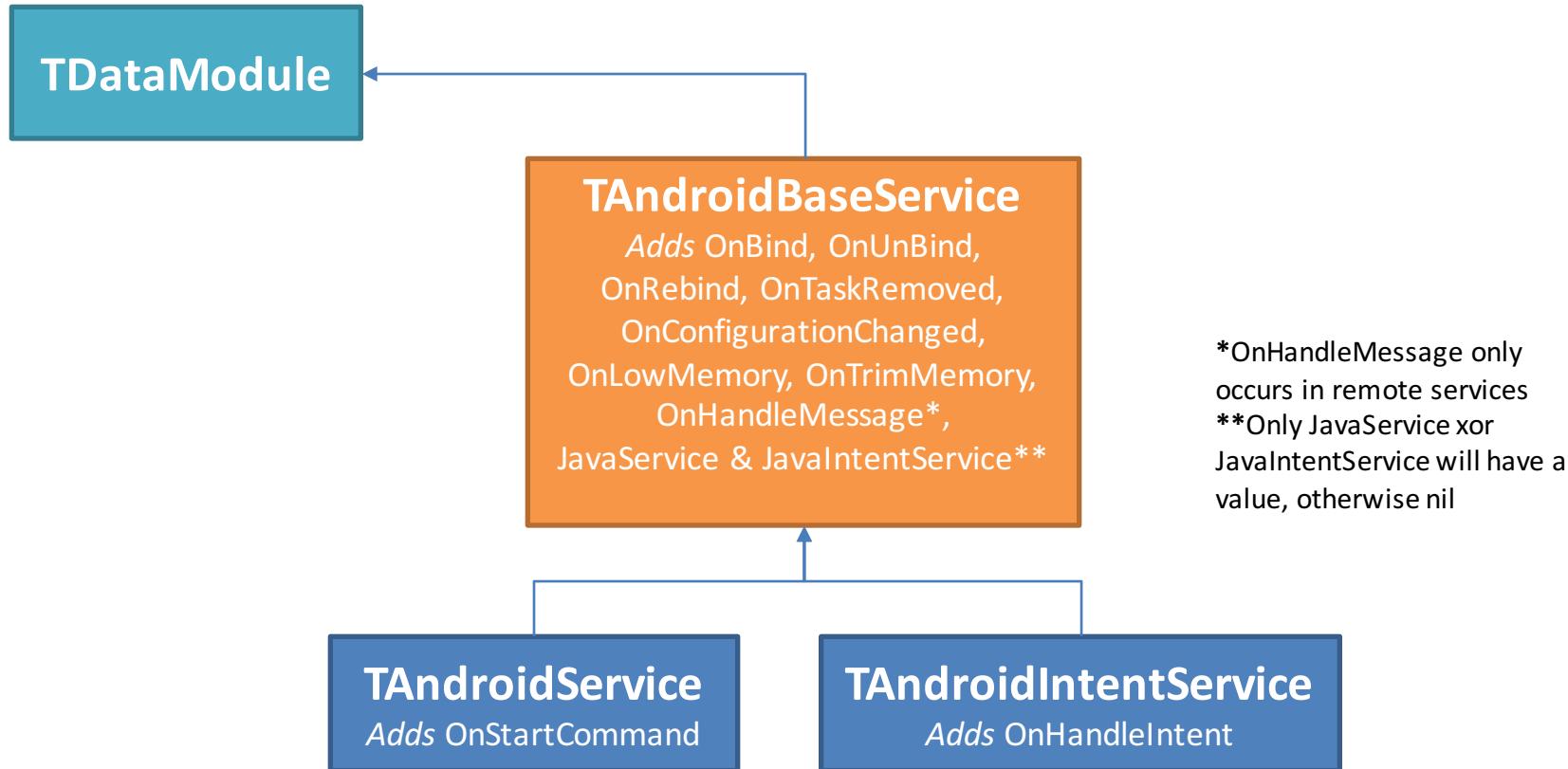
- IntentService
 - Handle asynchronous requests (onHandleIntent(Intent) event)
 - Otherwise are synchronous (onStartCommand(Intent) event)
 - Both are Bindable (onBind event)
- Local vs. Remote
 - Local only accessible by host (exported = false in manifest)
 - Remote cross app access (exported = true in manifest)

Android Service Wizard

- Local Service
 - Synchronous local access
- Intent Local Service
 - Asynchronous local access
- Remote Service
 - Synchronous local & remote access
- Intent Remote Service
 - Asynchronous local & remote access



Service Class Hierarchy



Services in Delphi

- Separate project
- Referenced by, and included in host app
- Must build service before adding to host
- For best results put projects in own directories

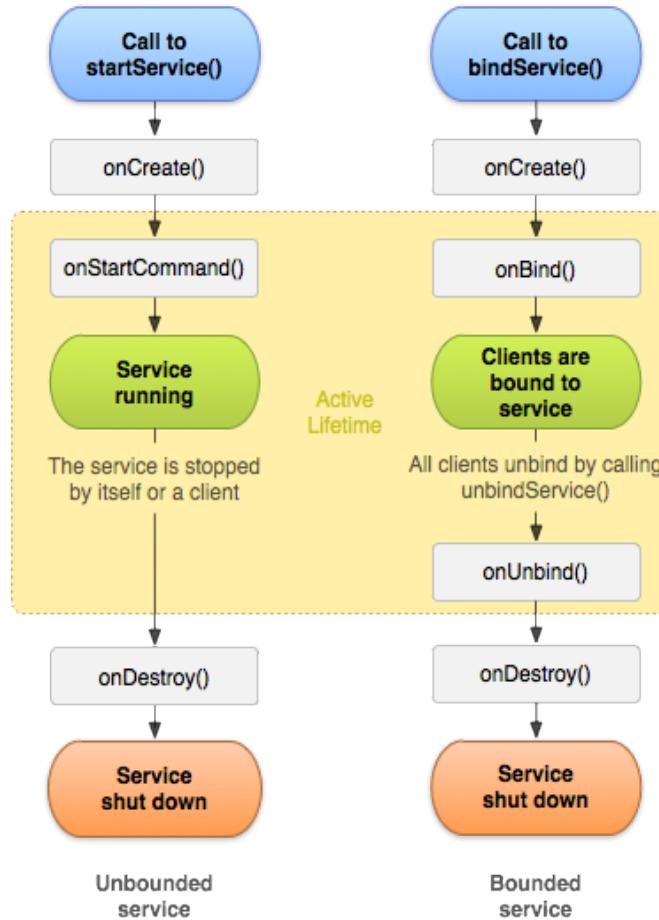
How to use a Service

- We have two ways to use a Service:
 1. Sending a request (`startService(Intent)`)

It is used to execute an unattended work in background
return `START_STICKY` or `START_REDELIVER_INTENT` in `OnStartCommand` to restart the service in case the app is closed
 2. Bound to the service (`bindService`)

It is used to create a long-standing connection to interact with the service

Service lifecycle



Restrictions for Services

- Cannot Access UI
- Not use components from the FMX namespace
- Be aware of power consumption
 - Using a lot of CPU
 - Using network connection
- Respect users expectations and privacy

Service Workarounds

- Use RTL or low-level calls instead of FMX components. If the service fails when started, use a lower level.
- Manually edit manifest to make remote services published.
- Manually edit Java Templates for Intent and Remote Services.

Android Services

Demonstrations

Summary

- Android services allow for execution even when app is not running.
- Add a new Android Service project to your app.
- Build service before adding reference.
- More information:
 - <http://delphi.org/tag/android-services/>
 - [http://embt.co/creating android services](http://embt.co/creating_android_services)
 - [http://embt.co/android service](http://embt.co/android_service)