Esri International Developer Summit

Palm Springs, CA

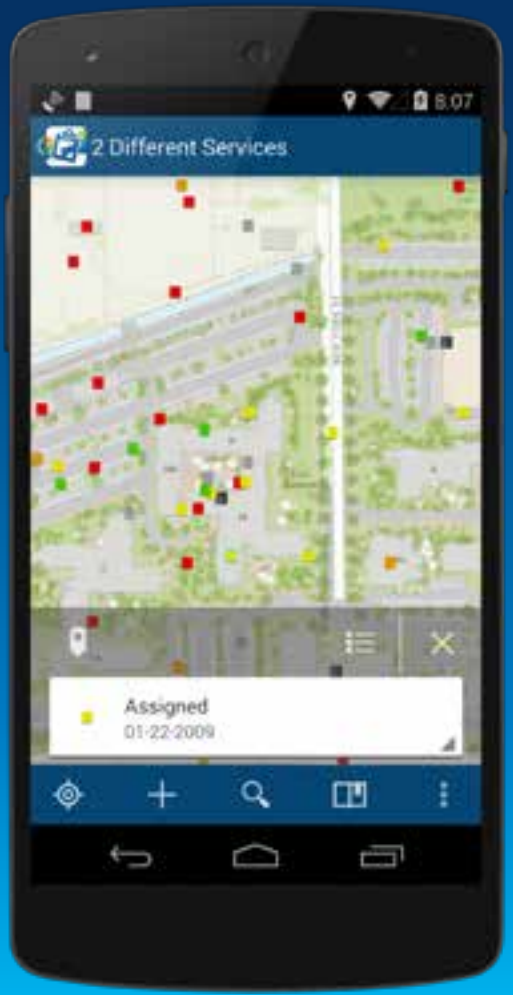
Collector for ArcGIS - How We Did It

Marc Bernstein, Eric Ito

Who are these guys?

How do we make Collector?


Carrier
12:28 AM
Maps
More
Assigned
01-22-2009


8:07
2 Different Services
Assigned
01-22-2009

Agenda

- Collector’s controller code
- Online/offline web maps
- Online/offline features on Android
- Background Processing on iOS

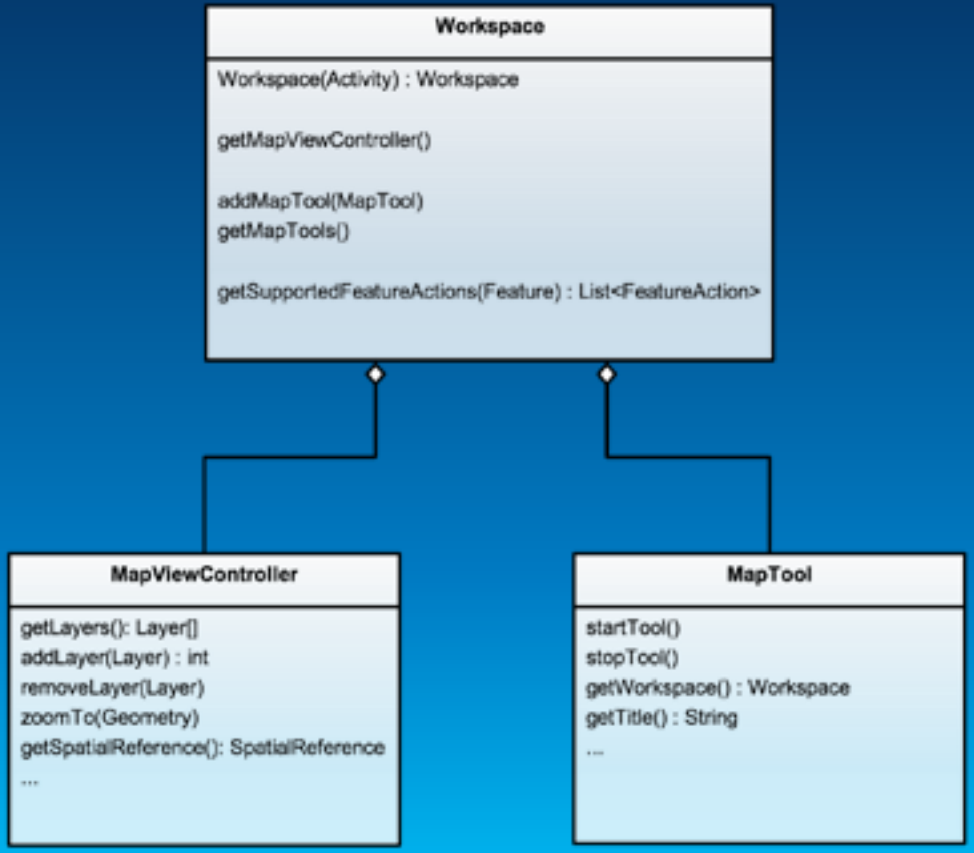
Collector’s controller code


Workspace
Map
MapTools
Results Panel
Feature Actions
Maps
More
Assigned
01-22-2009
2 Different Services
Assigned
01-22-2009

Tying everything together

- Workspace
- MapViewController
- MapTools
- FeatureActions

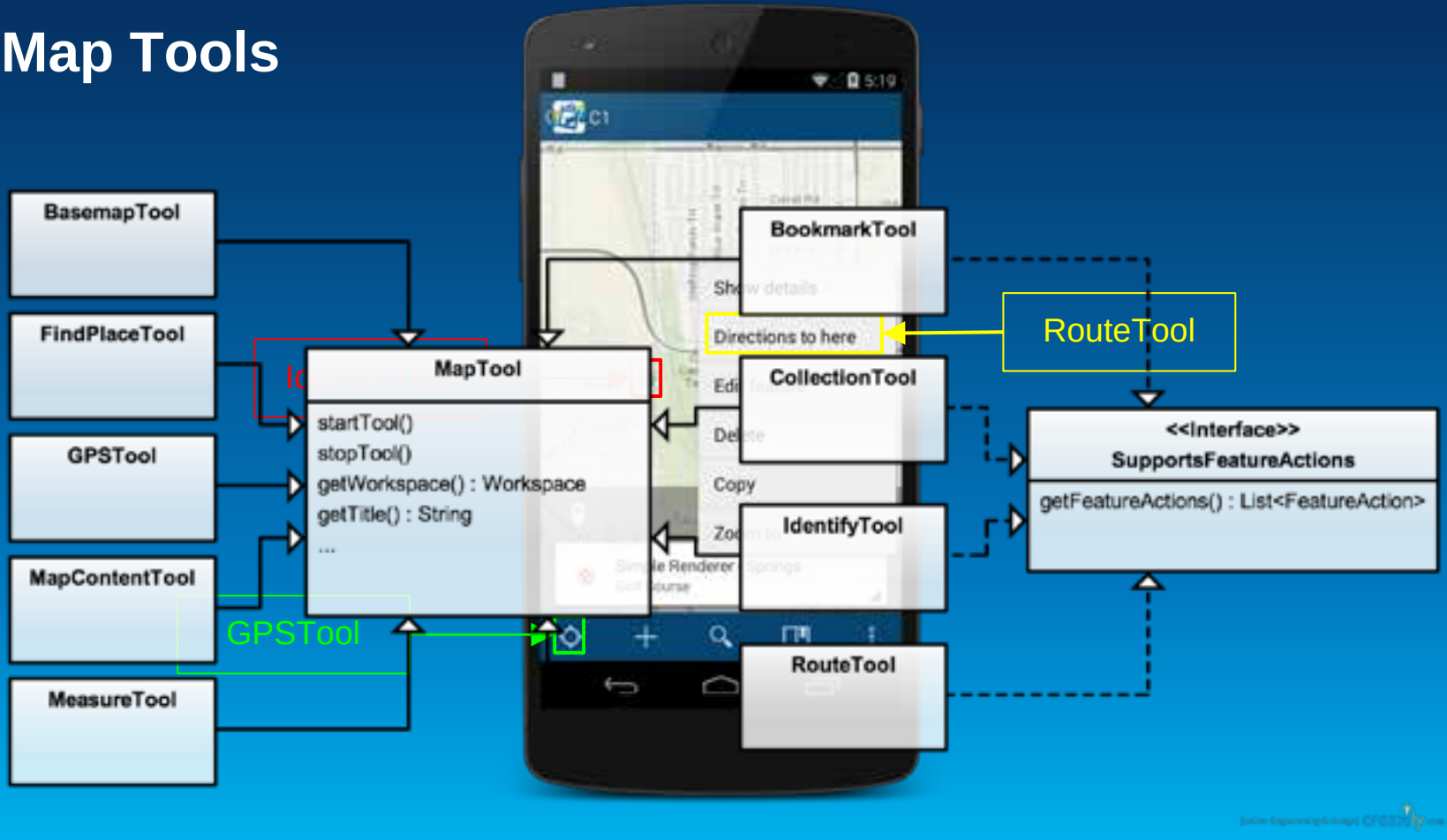
Workspace


Workspace
Workspace(Activity) : Workspace
getMapViewController()
addMapTool(MapTool)
getMapTools()
getSupportedFeatureActions(Feature) : List<FeatureAction>
MapViewController
getLayers(): Layer[]
addLayer(Layer) : int
removeLayer(Layer)
zoomTo(Geometry)
getSpatialReference(): SpatialReference
...
MapTool
startTool()
stopTool()
getWorkspace() : Workspace
getTitle() : String
...

Workspace - Example

```
Workspace workspace = new CollectorWorkspace(activity, listener);

// mapFragment implements MapViewController interface + contains MapView
workspace.setMapViewController(mapFragment);

MeasureTool measureTool = new MeasureTool(activity, workspace);
workspace.addTool(measureTool);

GpsTool gpsTool = new GpsTool(activity, workspace);
workspace.setGPSController(gpsTool);
workspace.addTool(gpsTool);

// Add CollectionTool, FindPlaceTool, BookmarkTool, MeasureTool, etc.
```

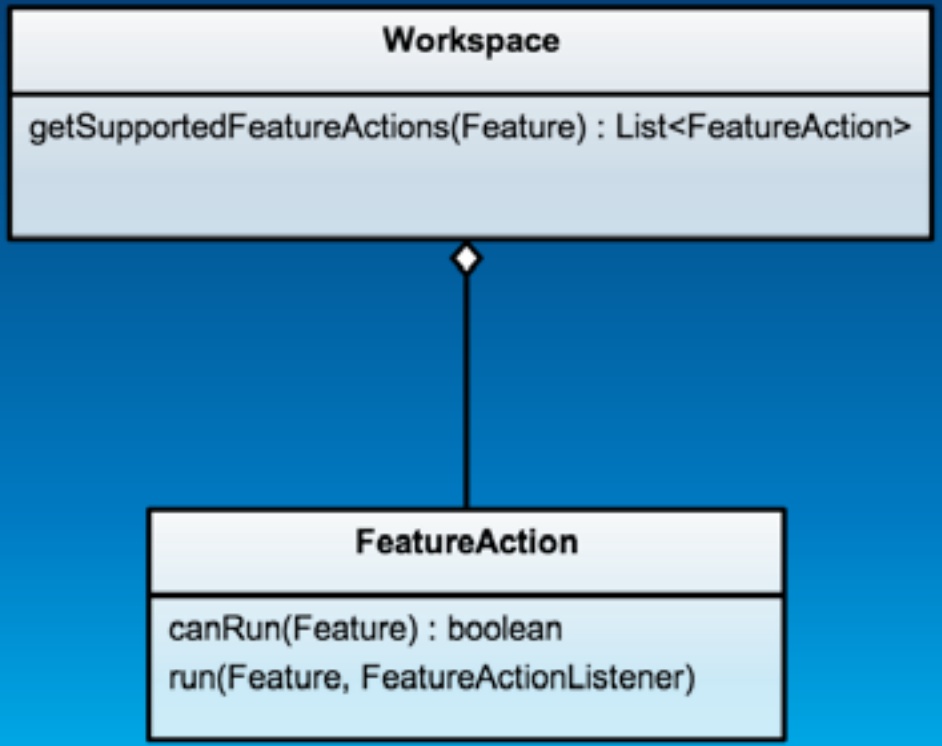
Map Tools


BasemapTool
FindPlaceTool
GPSTool
MapContentTool
MeasureTool
MapTool
startTool()
stopTool()
getWorkspace() : Workspace
getTitle() : String
...
GPSTool
BookmarkTool
Directions to here
CollectionTool
Copy
IdentifyTool
RouteTool
RouteTool
<<Interface>>
SupportsFeatureActions
getFeatureActions() : List<FeatureAction>

Map Tool - Example

```
public class CollectionTool extends MapTool {

  public CollectionTool(Activity activity, Workspace workspace) {
    super(activity, workspace);
    // Set flux capacitor to 1.21 gigawatts
  }

  @Override
  public void startTool() {
    super.startTool();
    // Initialize, set up UI if needed
  }

  @Override
  public void stopTool() {
    super.stopTool();
    // Break down UI, cleanup
  }

  public void startEditing(CollectionMode mode, Feature feature) {
    // Step 1. Edit your feature
    // Step 2. ???
    // Step 3. PROFIT!!!
  }
}
```

Feature Actions



Feature Action - Example

```
@Override
public List<FeatureAction> getFeatureActions() {
  List<FeatureAction> featureActions = new ArrayList<FeatureAction>();

  featureActions.add(new FeatureAction("Edit Feature", new FeatureActionCodeBlock() {

    @Override
    public boolean canRun(Feature feature) {
      return isEditable(featureLayer, feature);
    }

    @Override
    public void run(Feature feature) {
      startEditing(CollectionMode.EDIT, feature);
    }
  }));

  return featureActions;
}
```

Handling Online/Offline Maps

Handling Online/Offline Maps

- Previously Collector only dealt with Portal Items (web map)
- Needed an abstraction so controller code could deal with a common model object

Map Item

- Acts as a proxy between controller and the online/offline items
- Controllers only know about Map Item
- Internally points to either an online/offline map

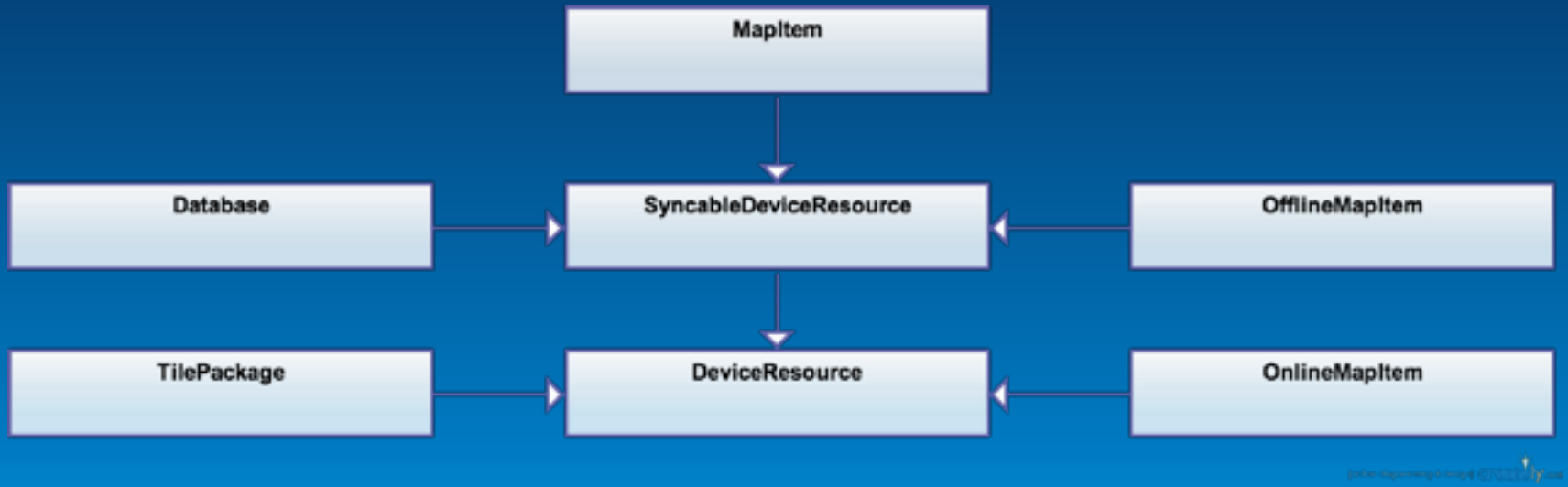
Online Map Item

- Portal Item backed by a web map
- Connected layers

Offline Map Item

- Collection of “on device” resources
- Tile Packages (LocalTiledLayer)
- Geodatabases (FeatureTableLayer)

MapItem-related objects



Downloading a MapItem

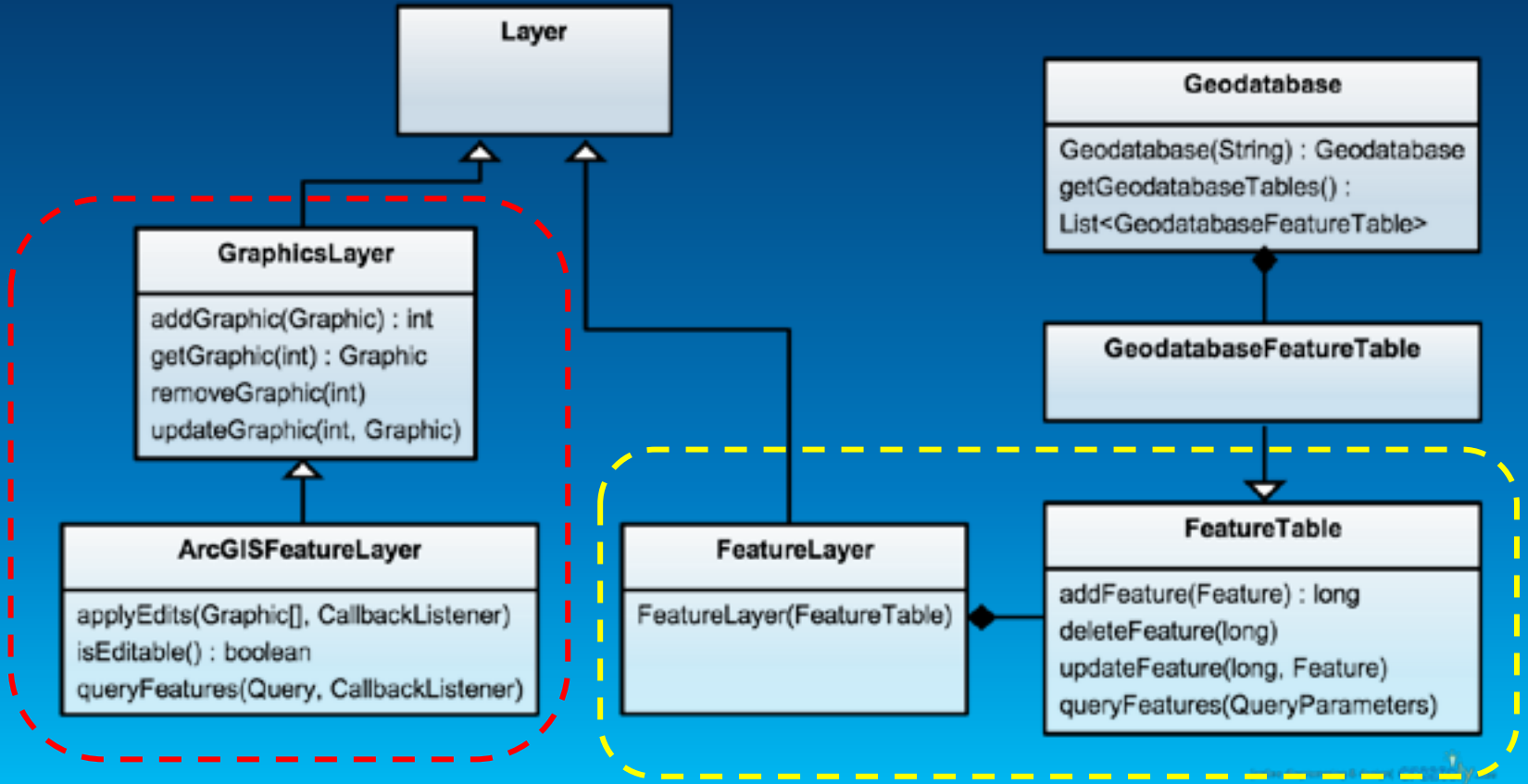
```
// kickoff the download a map item
[mapItem downloadArea:downloadInfo
                       toPath:fullPath
                 withProgress:nil
               withCompletion:[weakSelf downloadCompletionBlockForMapItem:mapItem]];

// MapItem implementation

// loop through this MapItem's resources and call
// download on each of them...
for (DeviceResource *resource in deviceResources) {
  [resource downloadArea:downloadInfo
                  toPath:self.path
            withProgress:progressBlock
              completion:completionBlock];
}
```

Online/offline features on Android

All Features are not created equal… (for now)

- Android specific issue at 10.2.2
- ArcGISFeatureLayer and FeatureLayer don’t share interface
- Particular challenge for the 10.2.2 release on Android

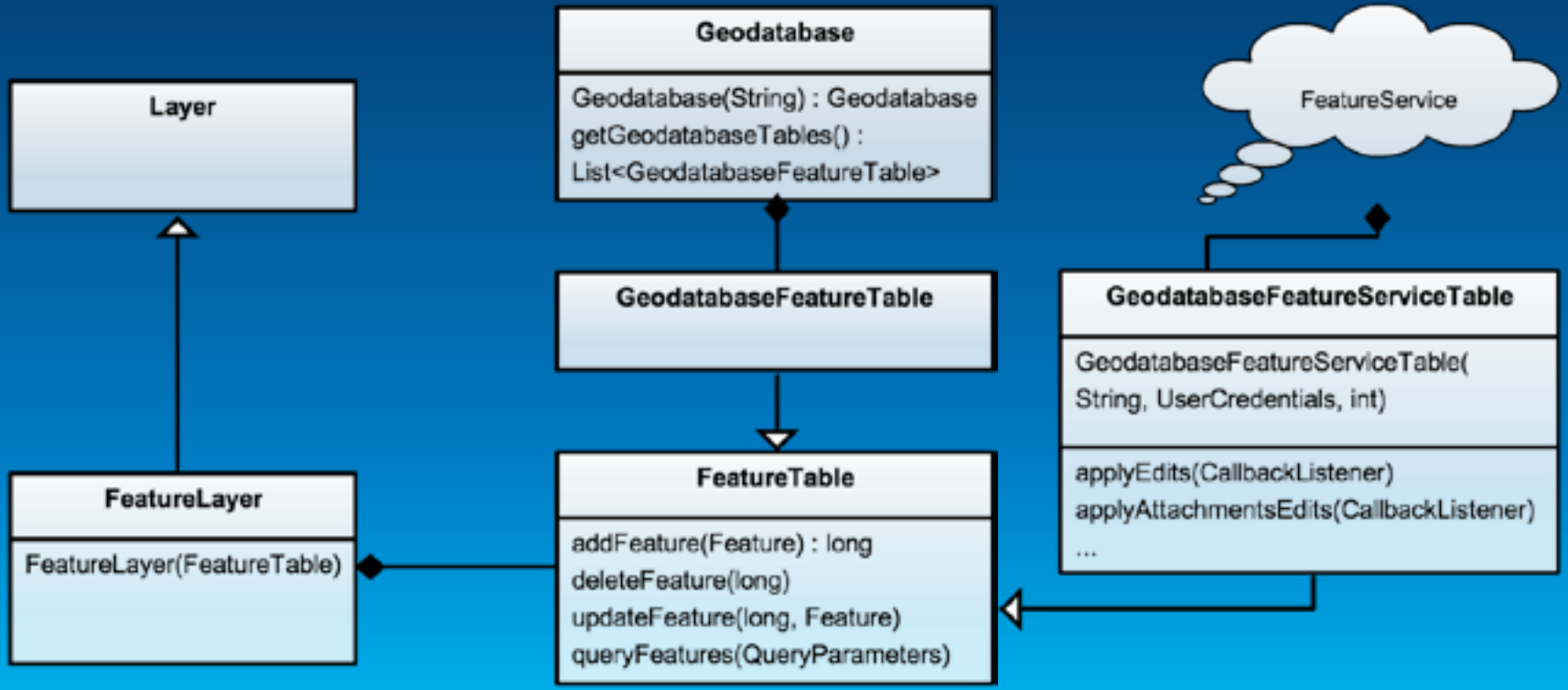
Feature Layer & ArcGISFeatureLayer at 10.2.2



FeatureItem example

```
public class FeatureItem {
  private Layer mLayer;

  private Feature mFeature;

  public Feature getFeature() {
    return mFeature;
  }

  public Graphic getGraphic() {
    if (mFeature instanceof Graphic) {
      return (Graphic) mFeature;
    }
    return new Graphic(mFeature.getGeometry(), mFeature.getSymbol(), mFeature.getAttributes());
  }

  public Symbol getSymbol() {
    if (mLayer instanceof FeatureLayer) {
      return ((FeatureLayer) featureLayer).getRenderer().getSymbol(mFeature);
    } else if (mLayer instanceof ArcGISFeatureLayer) {
      return ((ArcGISFeatureLayer) featureLayer).getRenderer().getSymbol(mFeature);
    }
    return null;
  }
}
```

Feature Layer model, next release

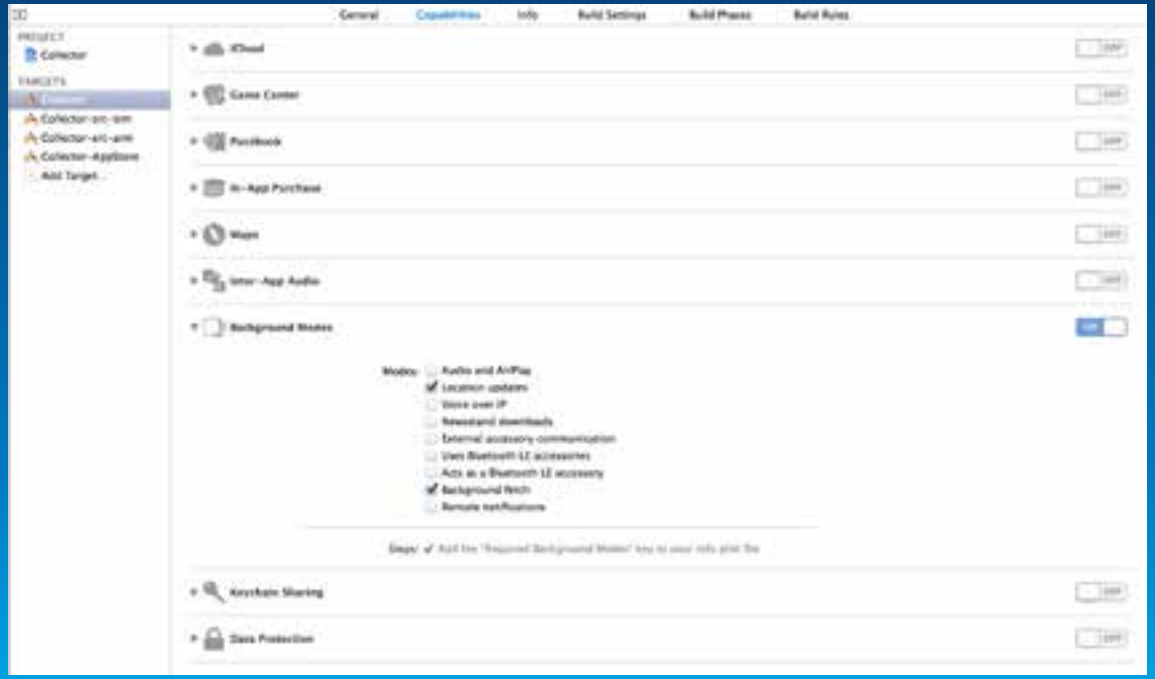


Background Processing on iOS

Background Processing (iOS)

- iOS 7 limits background tasks to 3 minutes
- Downloading a map may take longer (exporting tiles, generating a geodatabase)
- How can we check status in the background?

Background Fetch



Background Fetch (step 1)

```
// set background fetch interval so our app is called to update status
[[UIApplication sharedApplication] setMinimumBackgroundFetchInterval:
 UIApplicationBackgroundFetchIntervalMinimum];
```

Background Fetch (step 2 - check status)

```
#pragma mark Handle Background Fetch to check status

-(void)application:(UIApplication *)application performFetchWithCompletionHandler:(void (^)
   (UIBackgroundFetchResult))completionHandler
{
    //
    // check if we have any active resumeIDs
    if ([[[AGSTask activeResumeIDs] allKeys] count]) {
        //
        // this will trigger status checks for any active jobs. If a job is done
        // and a download is available, a download will be kicked off
        [AGSTask checkStatusForAllResumableTaskJobsWithCompletion:completionHandler];
    }
    else {
        //
        // we should call this right away so the OS sees us as a good citizen.
        completionHandler(UIBackgroundFetchResultNoData);
    }
}
```

Downloading in the Background

```
#pragma mark Handle Background Downloads

- (void)application:(UIApplication *)application handleEventsForBackgroundURLSession:(NSString *)
   identifier completionHandler:(void (^)())completionHandler
{
    [[AGSURLSessionManager sharedManager] setBackgroundURLSessionCompletionHandler:completionHandler
                                                                        forIdentifier:identifier];
}
```

Resuming Downloads

- App can be killed by the user explicitly
- Evicted by iOS if the system needs resources

Resuming Map Downloads

```
//Automatically kick off download of any maps that haven't fully downloaded for this account
NSArray* accountMaps = [[MTFOnDeviceMaps shared] allMapsForAccount:account];
for (MTFMapItem* offlineMap in accountMaps){
    if (offlineMap.status == MTFOnDeviceDownloading) {
        [offlineMap resumeDownloadWithProgress:nil
                                withCompletion:[weakSelf downloadCompletionBlockForMapItem:
                offlineMap]];
    }
}
```

Resuming Resource Downloads

```
//
// if we have a resumeID, we can just resume the previous job
if (self.resumeID.length) {
    self.gdbOperation = [self.gdbSyncTask generateGeodatabaseWithResumeID:self.resumeID
                                                                   status:statusBlock
                                                               completion:completionBlock];
}
else {
    self.gdbOperation = [self.gdbSyncTask generateGeodatabaseWithParameters:params
                                                            downloadFolderPath:path
                                                                   useExisting:YES
                                                                        status:statusBlock
                                                                    completion:completionBlock];
}

//
// save our resumeID so we can resume if necessary
self.resumeID = self.gdbOperation.resumeID;
```

Other sessions of interest

Building Offline Apps for iOS and the Mac

- **Wed 2:30pm - 3:30pm**

Building Android Apps with ArcGIS Runtime SDK

- **Wed 2:30pm - 3:30pm, Thu 2:30pm - 3:30pm**

Building iOS Apps with ArcGIS Runtime SDK

- **Wed 10:30am - 11:30am, Thu 1:00pm - 2:00pm**

20 Things You Didn't Know You Can Do with ArcGIS Runtime SDK for iOS

- **Wed 2:30pm - 3:00pm**

Building Offline Apps with ArcGIS Runtime SDKs—Part I & II

- **Wed 4:00pm - 5:00pm & Wed 5:30pm - 6:30pm**

Thanks for attending!

- http://www.esri.com/events/devsummit/session-rater
- Search “Collector for ArcGIS—How We Did It”
- Come see us at the Mobile Islands in the Showcase

Understanding our world.