Places in code to touch when adding new World Objects to MapSDK and GeodataService.

in MapSDK (com.reconinstruments.mapsdk)

|  |  |
| --- | --- |
| geodataservice.clientinterface.worldobjects | add new WO to e**um** WorldObjectTypes |
|  | create new *object*.java. I suggest copying an existing object type and modify. Three types: POI, Terrain and Paths (Roads/Trails) |
| geodataservice.clientinterface | create .aidl file for new class. Look at others for examples. |
| geodataservice.clientinterface.GeoTile.java | import new class  add handling in *readFromParcel(Parcel \_parcel)* |
| new rendering parameters for type | add new type to *enum* at top of mapview.WO\_drawings. RenderingSchemeManager.java . If needed, mod code to read in new parameters and serve them through the API |
| storageDependencies.../ rendering\_schemes.xml | add new xml parameters as needed |
| in WorldObjectDrawing | add class name to enum WorldObjectDrawingTypes  add case to GetCorrespondingWorldObjectType  add case to ClassForDrawingType  add case to DrawingTypeForString |
| in mapview.WO\_drawings.GeoTileDrawing | add 2 cases for creation |
| in POIDrawing, TerrainDrawing OR TrailDrawing | add case in creator |
| in map definition files | add appropriate object layers for new classes |

In Geodata service

|  |  |
| --- | --- |
| in data source, eg, datasourcemanager.Recon\_Data.ReconBaseDataSource | add ***enum*** *ReconBaseDataTypes*  add case to *loadRGZTile()* to handle new type |
| in transcoder, eg datasourcemanager.Recon\_Data.ReconBaseTranscoder | add class to lookup tables in *init()*  add handling code in *TranslateDataRecordIntoWO()*- add case for new obj  add include for new class |
|  |  |
|  |  |

**at end,**

**cd MapSDK**

**ant clean**

**bash make\_app.bash and confirm no issues**

**cd ../GeodataService/GeodateService**

**bash clean\_build\_<etc>.bash and confirm no issues**

**cd to map app of choice and build**