

# 基于 FME 的矢量地理底图数据分幅及栅格化

武汉市测绘研究院 张敏

## 1 背景

在日常工作中，我们有时会遇到需要将大范围内多要素的地理底图数据根据特定的要求分幅裁切的情况，并要求各要素保持原有的各种几何、属性信息。现有此类情况，需要将武汉市全市域范围内 MDB 格式的整张多要素地理底图数据裁切为 1:2000 标准分幅的 MDB 文件，并配置线型、符号、注记等对其进行栅格化，输出为带坐标的图片文件，并制作瓦片数据导入移动终端供外业调查使用。

## 2 数据概况

### 2.1 全市域多要素 mdb 矢量数据

主要数据源为武汉市全市域范围的地理地图 MDB 格式矢量数据，包括有：地名、城市道路、公路、轨道交通、铁路、建筑、大型市政设施、水体、山体、绿地等类别，其中每个大类中又包含若干个小要素类，比如城市道路分为城市主干路、城市其他道路等。这些矢量数据中，都包含各自必须的属性信息，后面出图时可通过这些属性来生成文字标注。

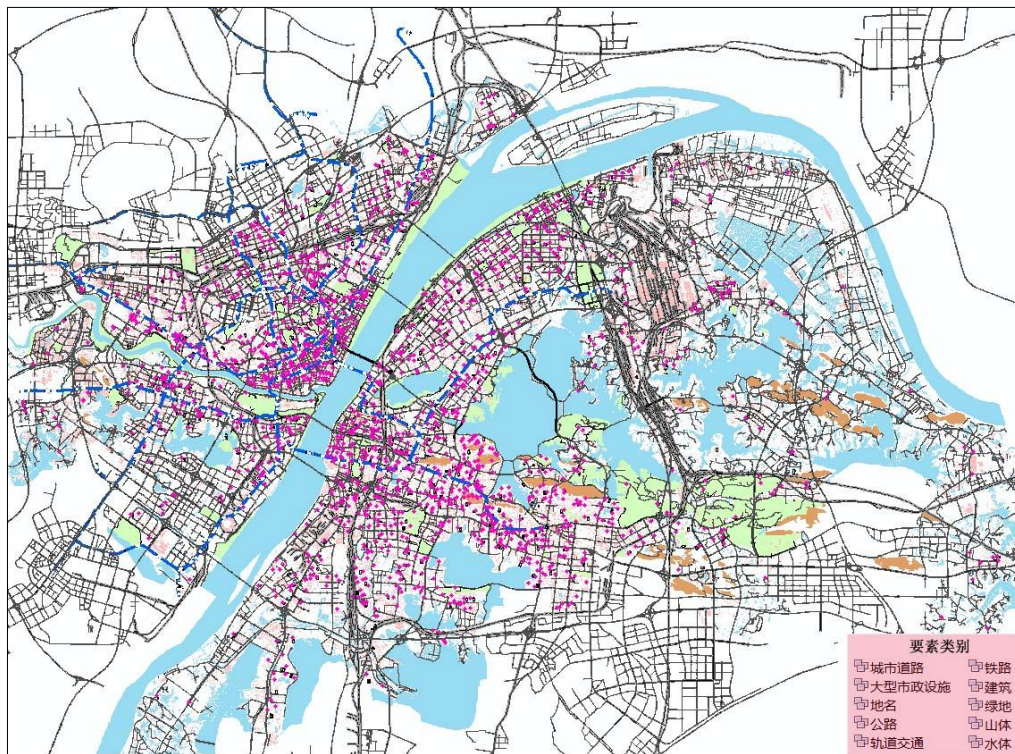


图 2.1 全市域地理地图要素（局部）

### 2.2 全市域 1:2000 标准分幅格网

为了将整张的全市域范围的地理底图进行分幅裁切，需要借助全市域的 1:2000 标准分幅格网 shapefile 数据来完成。每个格网的属性信息中都包含有对应位置标准分幅文件的文件名，即 *filename* 字段的值。进行分幅文件输出的时候，需要将裁切后的要素按照对应的位置关系拆分到对应的标准分幅 2000 图中，并以规定的文件名来命名保存（1:2000 比例尺地形图数据文件名为 8 位，前 6 位分别取图幅坐标三位整公里数，X 坐标在前，Y 坐标在后）。

第 7、8 两位为 0，如：39782100.mdb)。

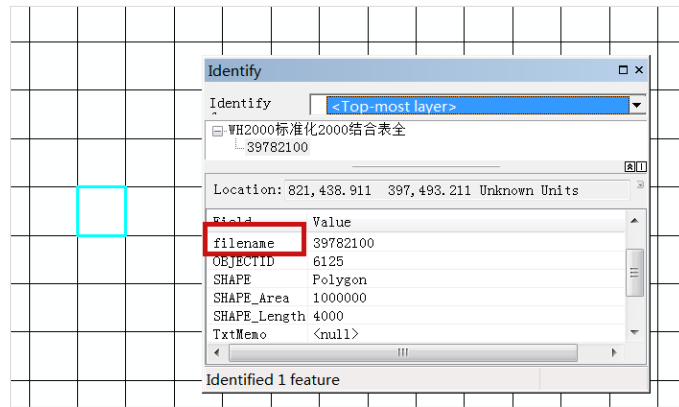


图 2.2 1:2000 标准分幅格网

### 3 FME 实现

#### 3.1 基本思路

##### 3.1.1 分幅裁切矢量地理底图数据

运用 *Clipper* 转换器，利用标准分幅格网矢量数据，对整张的地理地图矢量文件进行裁切，并保存为 1:2000 标准分幅的多个 MDB 文件。这些分幅文件，后期将被用于数据处理、栅格化。

##### 3.1.2 配置要素符号样式

运用 *MapnikRasterizer* 转换器将矢量数据输出成栅格图像。在输出之前，需要对各要素的符号、线型、颜色、注记等进行配置，以使输出的图片能满足工作需要。需要注意的是，对于需要生成注记的要素，其设置的属性字段名要与其矢量数据中完全一致，否则无法按要求输出文字标注。

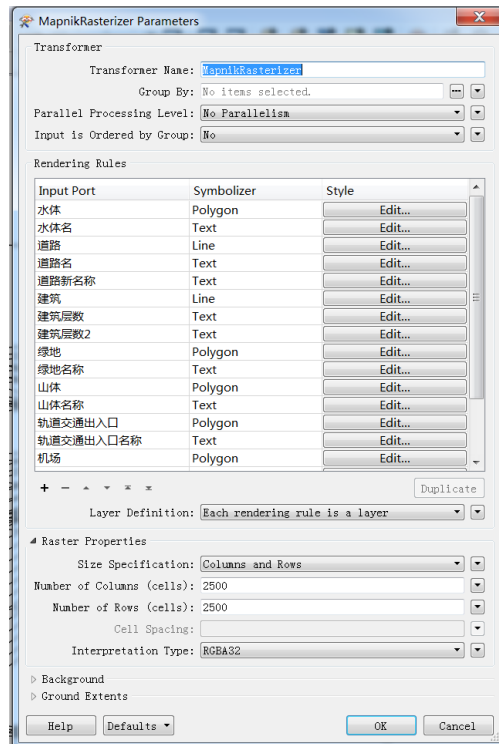


图 3.1 栅格化样式参数配置

### 3.1.3 关联格网与分幅底图

由于裁切后的分幅 MDB 底图文件中的要素类型、范围等会有所不同，而输出的图片尺寸是固定统一的，就会导致相邻的图片中的要素无法正常接上。如下，右侧图幅中要素如果未能布满整个图幅范围，那么图片输出的时候就会自动放大使其布满整个图片范围，但左图中的要素都充满了幅图，所有导致两张图片中的要素缩放比例不一致：

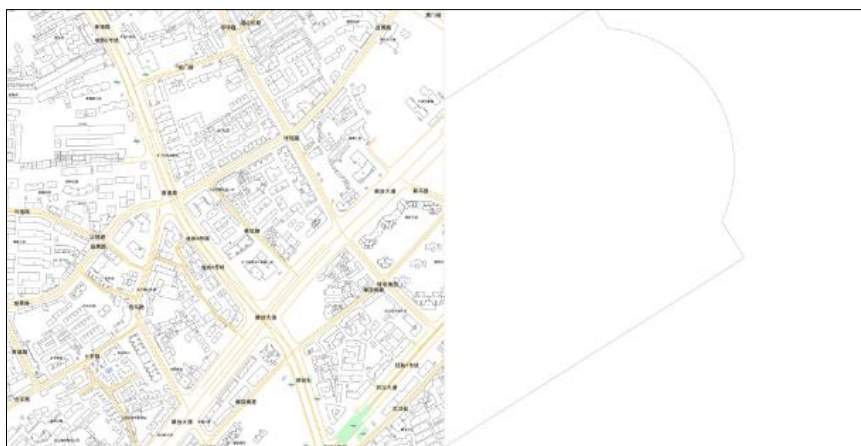


图 3.2 图幅未接边示例

为了解决这一问题，需要将每幅图对应的格网一起输出，这样每幅分幅底图中的要素范围都一样，输出要素的缩放比例也一致，因此接边图幅中的要素都能正确接边。如何才能从全市域范围的格网中选中与某图幅匹配的格网呢？可以根据分幅 MDB 文件名称，批量利用 **where** 语句在格网中进行查找匹配。先根据每个读取进来的分幅文件名来创建一个新的属性（假设为 *a*），使其属性值等于该文件名。新增一个 *FeatureReader* 转换器来读取格网数据，在其读取的 **where** 条件语句中规定读取格网中 *filename* 字段值与属性 *a* 的值相等的格网（前面提到过，格网数据的 *filename* 字段值代表的就是与该格网对应的标准分幅图的文件名），这样就能将与图幅对应的格网一起栅格化输出。

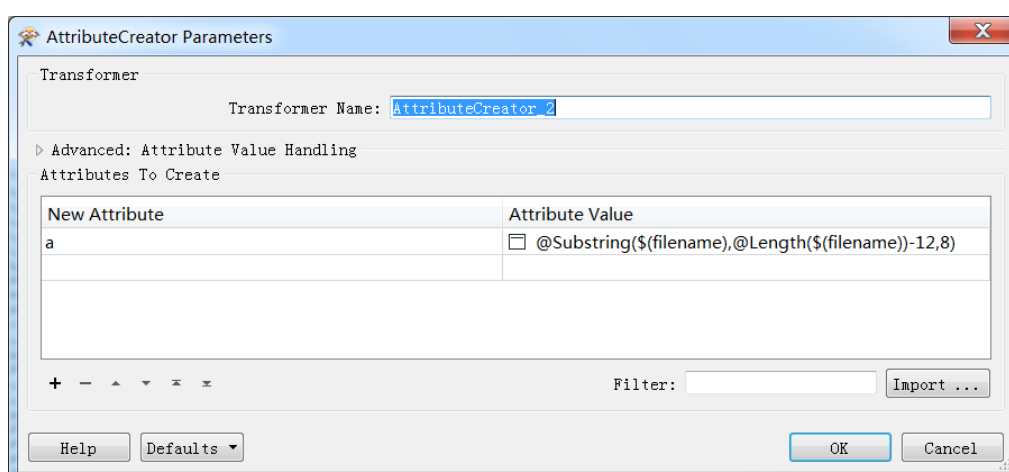


图 3.3 字段 a 定义

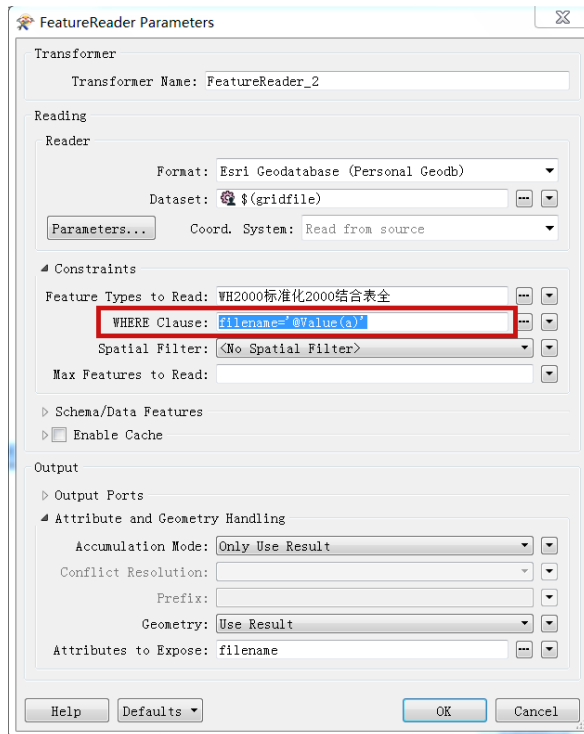


图 3.4 条件参数设置

### 3.1.4 循环执行栅格化

*MapnikRasterizer* 转换器每次只能输出一张图片,无法输出多张,所以需要添加 *Creator*、*WorkspaceRunner* 转换器来循环调用栅格化工作空间输出图片。栅格化的图片的命名,也需要与对应的 MDB 分幅底图保持一致,这可以通过与前面相同的方法(即创建新的属性)来实现,并将计算得到的属性值专递给 *Writer* 转换器来作为每幅栅格化图像的文件名。

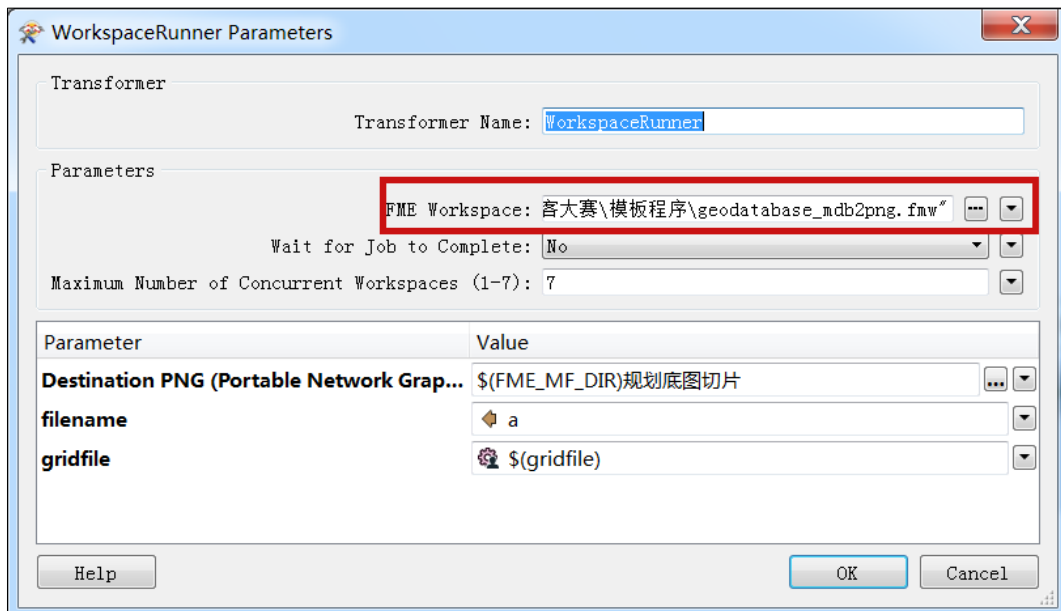


图 3.5 WorkspaceRunner 转换器设置

### 3.1.5 模板概览

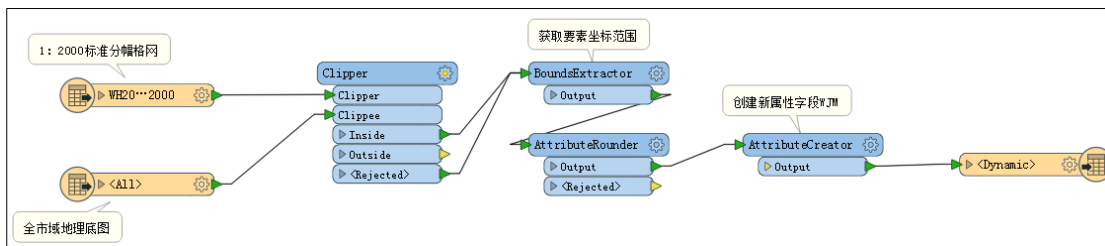


图 3.6 分幅裁切模板

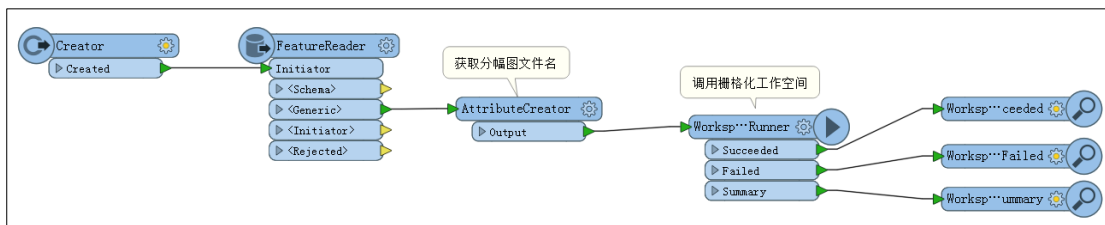


图 3.7 循环调用栅格化 workspace 模板

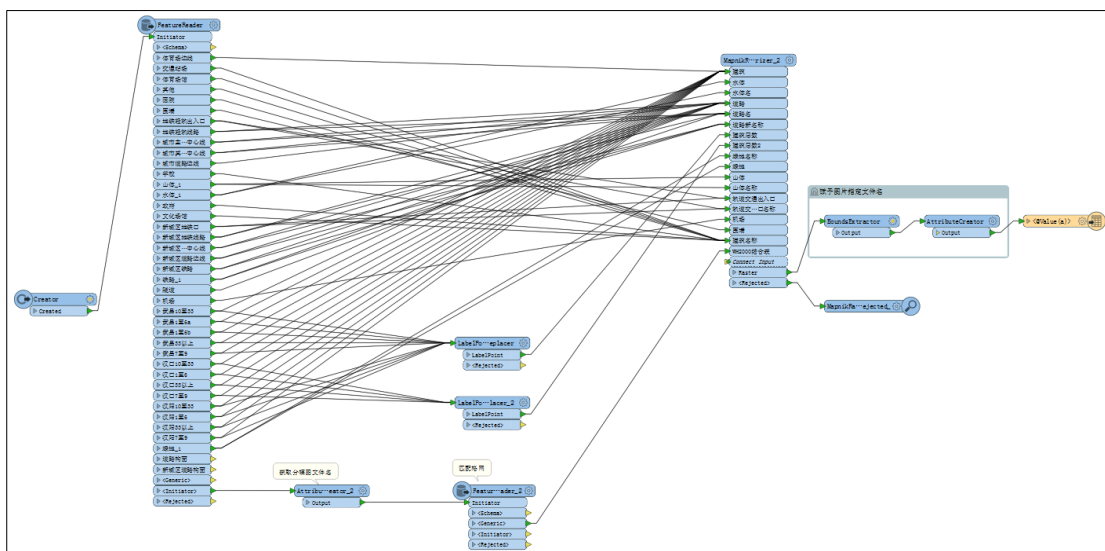


图 3.8 矢量数据栅格化模板

## 3.2 关键步骤

### 3.2.1 分幅输出设置

在利用标准分幅格网对地理地图数据进行裁切以后,如何才能将对应的要素输出到对应的单个分幅MDB文件中?扇出无疑是很好的解决方式。此处并没有使用 *SpatialFilter* 转换器对与格网相交的要素进行空间过滤,因为对于运算数据量很大的情况来说,这样执行空间操作效率较低。因此,此处避开了空间过滤,而是根据每个要素的属性字段值来设置扇出条件输出到特定图幅中。可以新建一个属性字段 *WJM* 来代表其所在图幅的文件名。由于每幅图的文件名都是根据其西南角坐标值来获得的,可根据此原理来利用各要素的坐标通过计算得到 *WJM* 字段的属性值。具体如下:通过 *BoundsExtractor* 转换器获取要素的最大最小坐标值,再利用 *AttributeCreator* 转换器来计算其所在图幅的文件名,属性值赋给 *WJM* 字段,最后在 *Writer* 的参数设置中,根据 *WJM* 字段的属性值进行扇出,那么文件名(*WJM* 字段值)相同的要素都会被保存到同一幅标准分幅的MDB文件中。

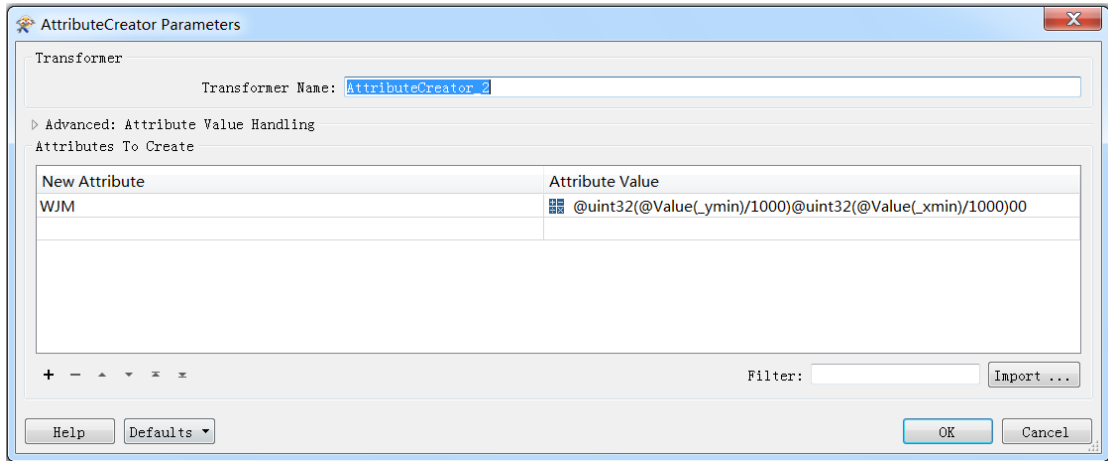


图 3.9 字段 WJM 定义

### 3.2.2 批处理转换器的使用

*WorkspaceRunner* 的功能是在本地计算机上调用另一个 FME workbench 工作空间文件生成新的 FME 进程，同时也能将转换器所在的工作空间中的某些值赋予被调用的工作空间。此转换器适用于批量处理，尤其是与目录和文件读模块一起使用。

由于有几千幅 1:2000 标准分幅的 MDB 地理底图文件需要进行栅格化，且格网文件中标准格网的数目比分幅图的数目多很多，所以必须设置条件，并采用批量处理转换器循环读取指定目录中的分幅数据，多次输出矢量数据栅格化的图片。每读取一个分幅地理底图文件，就触发 *WorkspaceRunner* 调用矢量数据栅格化的模板文件，使其生成该幅图的栅格图片。如此循环执行，直到读取完指定目录下最后一个分幅底图文件。

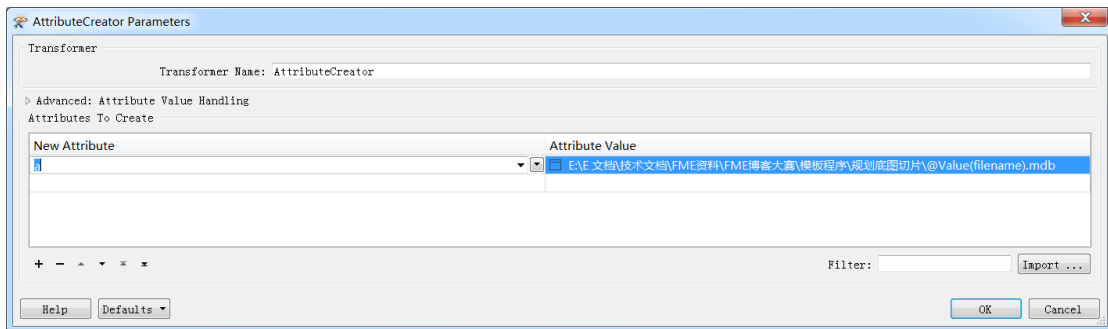


图 3.10 获取分幅底图文件名

### 3.2.3 发布参数设置

由于程序中有多处需要进行属性计算与传递的地方，要求有的参数必须发布出来供别的转换器使用，因此需要设置发布参数，保证这些参数能够在整个工作空间中进行传递与使用。

此外，有的属性如果在后面需要用到，在进行设置的时候，需要将其暴露（Exposed）出来。

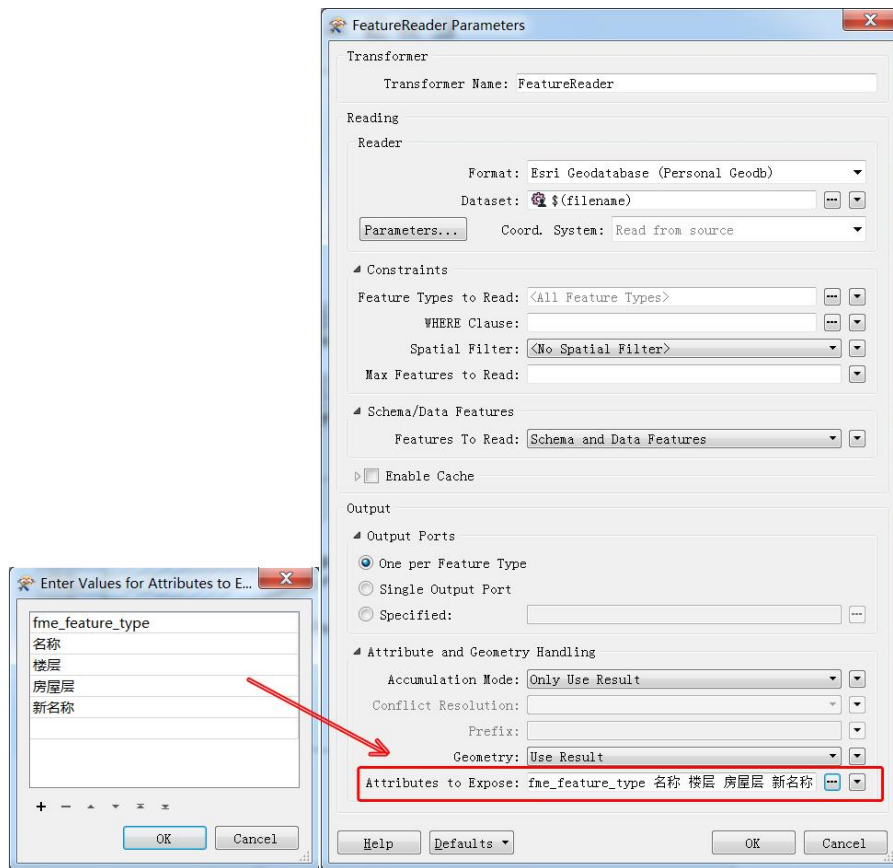


图 3.11 参数暴露设置

#### 4 总结

此次运用 FME 进行工作辅助，极大减轻了人力、节省了时间，也为后期工作中再次遇到类似需要批处理的情况提供了很好的借鉴意义。感谢同事白洁在此次工作中给予的帮助。

本人接触 FME 的时间并不长，但早就见识过它功能的强大，并深深为之折服。希望在今后的工作中，多多运用 FME 来解决问题，提高工作效率，不断提高自己在这方面的水平。