

使用 FME 实现农村土地承包经营权数据入库与质检

一、前言

上一次写 FME 在农村土地承包经营权确权中的应用，是两年前的这个时候，主要介绍了 FME 在农经权调查作业阶段的应用，如数据转换、属性挂接、公示表格的生成、地块四至的求取、地块示意图的生成等等。两年时间过去了，很多项目也已经到了验收阶段，验收很重要的一环，就是需要全县的数据全部通过农村土地承包经营权质检软件的检核。

农村土地情况十分复杂，地域差别较大，国标数据库也几经修改来满足各地出现的情况。有的地方有退耕还林、退耕还湖的，有的地方有集体流转的，为了适应本地的实际情况，各地都有相应的数据库及矢量数据的标准；有时候一个县是由多个标段分包的，作业单位不同，采购的软件可能不一样，软件如未能及时更新，数据结构也有些许区别，这就需要将多种数据整合起来，最终形成国标数据；数据质检是以区县为单位进行的，而实际调查过程中，均以行政村为单位进行，一个区县有几百个行政村，就需要将几百个行政村的数据，检查是否有压盖、重叠、不合理缝隙，修改的数据再合并成一个国标数据库。

FME 在数据并库中，应该扮演什么角色？是主角或是配角，是应该在前台作为主力军冲锋陷阵，还是在后台默默无闻帮某专业软件做做格式转换之类的龙套活？到底能发挥怎样的作用，怎样做能减少重复劳动，怎样做能提高数据质量。我觉得关键在于使用方法上，个人

认为，FME 不仅能在前台演好主角，还能很好地和其他软件配合，发挥出你意想不到的高效。

分析数据质检软件需要的数据可以看出，关键的数据有如下几项：

- 1) SHP 矢量数据（DK、JZX、JZD 等）
- 2) MDB 属性数据（FBF、CBF、DKXX、JTCY 等）
- 3) JPG 图件数据（含多页 PDF）
- 4) XLS 汇总数据（以权属单位统计的各类汇总表）

下面将结合本人使用 FME 处理这些数据的一些心得做个简要介绍。

二、 使用 FME 处理矢量数据

矢量数据最关键的数据就是 DK 图层，JZX、JZD 都可以从 DK 图层转出。所以 DK 图层的压盖、重叠等拓扑错误一定不能存在。并库时要将多个行政村、镇街数据合并成一个区县的数据，这样就会涉及到几种情况，一是有些地块在两个行政村都调查了，即 DK 合并后存在重叠；二是地块边线部分没有完全重合，存在犬牙交错的情况，即 DK 合并后存在压盖；三是地块边线虽然重合，但是节点不一致。这几种情况如果不在并库之前解决，到后期质检时再来处理会更麻烦。这里介绍一下我使用 FME 处理矢量数据的主要转换器和方法。

1) 重叠压盖检查---AreaOnAreaOverlayer

到了数据入库这个阶段，行政村内部的错误基本都已经解决了，但是并不排除不同行政村、镇之间完全不存在重叠或压盖。压盖检查

相对来说比较简单，使用 AreaOnAreaOverlayer 转换器即可。

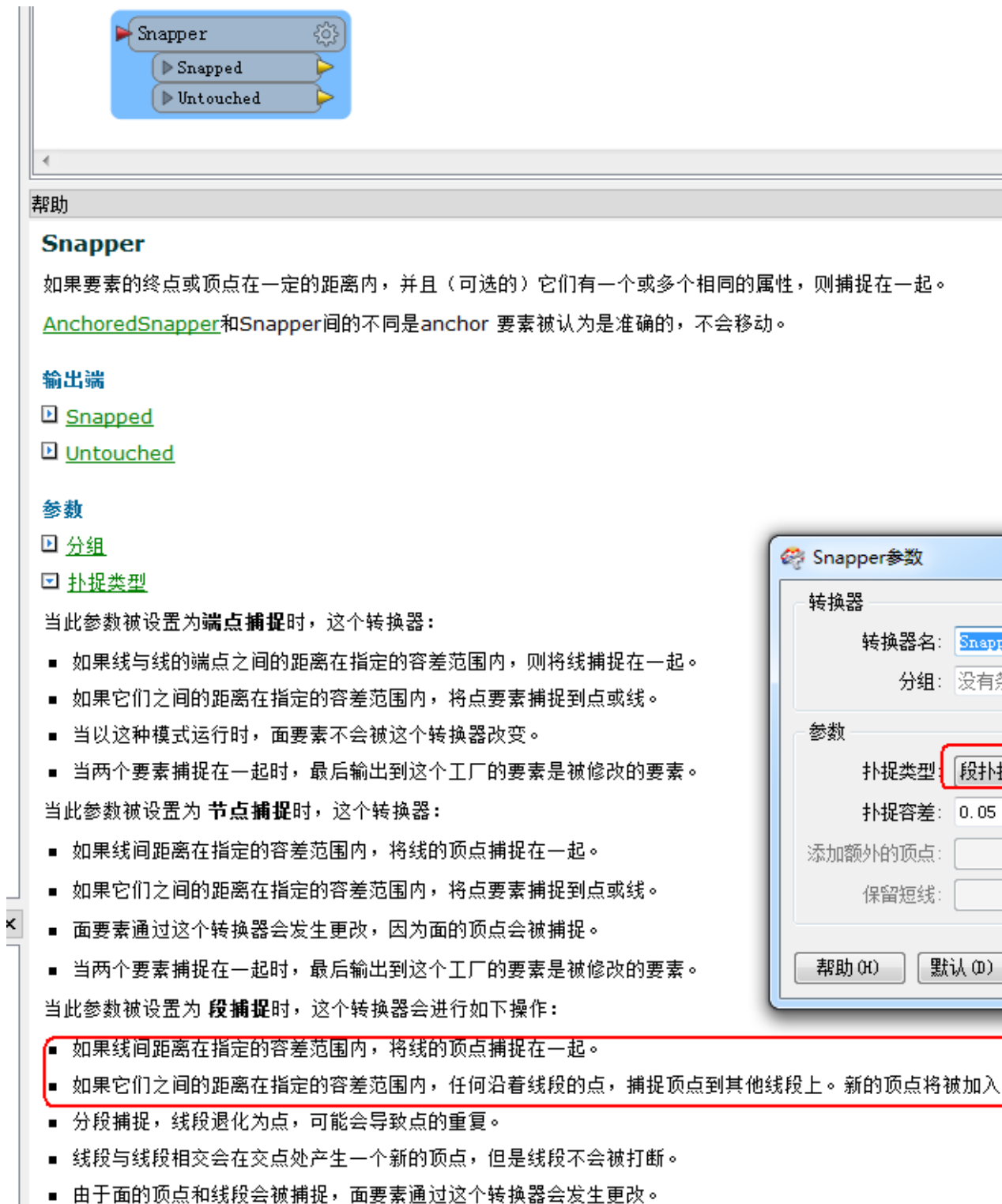


将面状图层输入到转换器后，判断输出端的 overlaps 值是否>1
即可判断是否存在压盖和重叠。

2) 数据接边的神器---Snapper 转换器

由于软件不同、精度不同，有些数据并不能算错误，只能算作误差。有的数据明明在 CAD 下已经接过边了，但是转换到 SHP 后，还

是发现有些地方存在很小的压盖，这种自然不能要作业员一个个来修改，对于这种误差的消除，使用 Snapper 转换器效果绝对让你满意。



The image displays the Snapper converter interface and its help documentation. At the top, a blue Snapper converter block is shown with two output ports: 'Snapped' and 'Untouched'. Below this, the '帮助' (Help) section for the Snapper converter is visible. It explains that the converter captures elements within a specified distance and optional attributes. It distinguishes between 'AnchoredSnapper' (where the anchor element is fixed) and the standard 'Snapper'. The '输出端' (Output) section lists 'Snapped' and 'Untouched'. The '参数' (Parameters) section includes '分组' (Group) and '捕捉类型' (Capture Type). The '捕捉类型' section describes three modes: '端点捕捉' (Endpoint Capture), '节点捕捉' (Node Capture), and '段捕捉' (Segment Capture). The '段捕捉' mode is highlighted with a red box, indicating it is the recommended setting for DK boundary lines. To the right, a 'Snapper参数' (Snapper Parameters) dialog box is shown, with the '捕捉类型' (Capture Type) set to '段捕捉' (Segment Capture) and '捕捉容差' (Capture Tolerance) set to 0.05.

Snapper

如果要素的终点或顶点在一定的距离内，并且（可选的）它们有一个或多个相同的属性，则捕捉在一起。
[AnchoredSnapper](#)和Snapper间的不同是anchor 要素被认为是准确的，不会移动。

输出端

- [Snapped](#)
- [Untouched](#)

参数

- [分组](#)
- [捕捉类型](#)

当此参数被设置为**端点捕捉**时，这个转换器：

- 如果线与线的端点之间的距离在指定的容差范围内，则将线捕捉在一起。
- 如果它们之间的距离在指定的容差范围内，将点要素捕捉到点或线。
- 当以这种模式运行时，面要素不会被这个转换器改变。
- 当两个要素捕捉在一起时，最后输出到这个工厂的要素是被修改的要素。

当此参数被设置为**节点捕捉**时，这个转换器：

- 如果线间距离在指定的容差范围内，将线的顶点捕捉在一起。
- 如果它们之间的距离在指定的容差范围内，将点要素捕捉到点或线。
- 面要素通过这个转换器会发生更改，因为面的顶点会被捕捉。
- 当两个要素捕捉在一起时，最后输出到这个工厂的要素是被修改的要素。

当此参数被设置为**段捕捉**时，这个转换器会进行如下操作：

- 如果线间距离在指定的容差范围内，将线的顶点捕捉在一起。
- 如果它们之间的距离在指定的容差范围内，任何沿着线段的点，捕捉顶点到其他线段上。新的顶点将被加入。
- 分段捕捉，线段退化为点，可能会导致点的重复。
- 线段与线段相交会在交点处产生一个新的顶点，但是线段不会被打断。
- 由于面的顶点和线段会被捕捉，面要素通过这个转换器会发生更改。

Snapper参数

转换器

转换器名: Snapper

分组: 没有分组

参数

捕捉类型: 段捕捉

捕捉容差: 0.05

添加额外的顶点:

保留短线:

帮助 (H) 默认 (D)

DK 边线的接边，应该使用段捕捉类型，这样，在容差范围内的线都会捕捉到一起并保持节点一致。这样处理过的图层，接边就不会

再出现拓扑问题了，使用面叠加判断是否有压盖错误，利用 Snapper 转换器消除容差内的误差，基本可以解决面图层的拓扑问题。但是这样会造成一个新的问题，如果我的界址点、线图层已经生成好了，现在 DK 图层修正了，JZD、JZX 与 DK 图层套和不上了该如何办，难道需要再转换一次或者手工挪到 DK 边线或者节点上？我们需要另外一个转换器 AnchoredSnapper（锚定捕捉）。

AnchoredSnapper

Anchor

Candidate

Snapped

Untouched

帮助

AnchoredSnapper

将一系列符合输入标准的要素，与位于特定容差范围内、也符合输入标准的要素进行捕捉。在转换时你可以用这个转换器来

AnchoredSnapper 和Snapper间的不同是节点要素被认为是准确的，因此不会移动。

输入端

Anchor

固定端

Candidate 要素可以捕捉到节点。弧将作为线性要素进行捕捉，椭圆将作为多边形要素进行捕捉。

Candidate

需要捕捉的端口

可以捕捉到Anchor要素的要素。弧将作为线性要素进行捕捉，椭圆将作为多边形要素进行捕捉。

输出端

Snapped

Untouched

参数

分组

捕捉类型

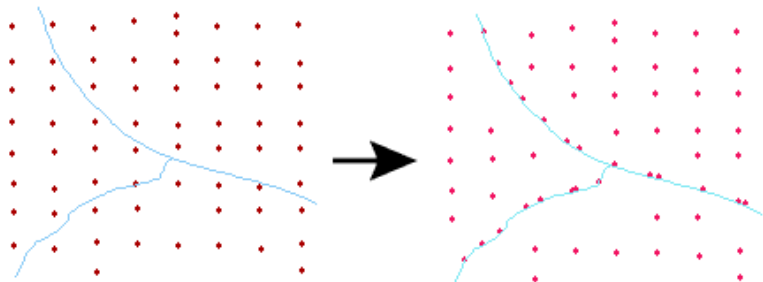
捕捉容差

添加额外的顶点

使用方法

- 任何经过捕捉导致维数压缩的要素会被记录为“退化”并删除。“维度压缩”指一条线或面变为一个点，或者一个面变成了-
- 在执行捕捉之后会有一个简短的清理步骤。这一步会移除重复的点，也可能会创建聚合要素来保持压盖的、有方向的线

举例



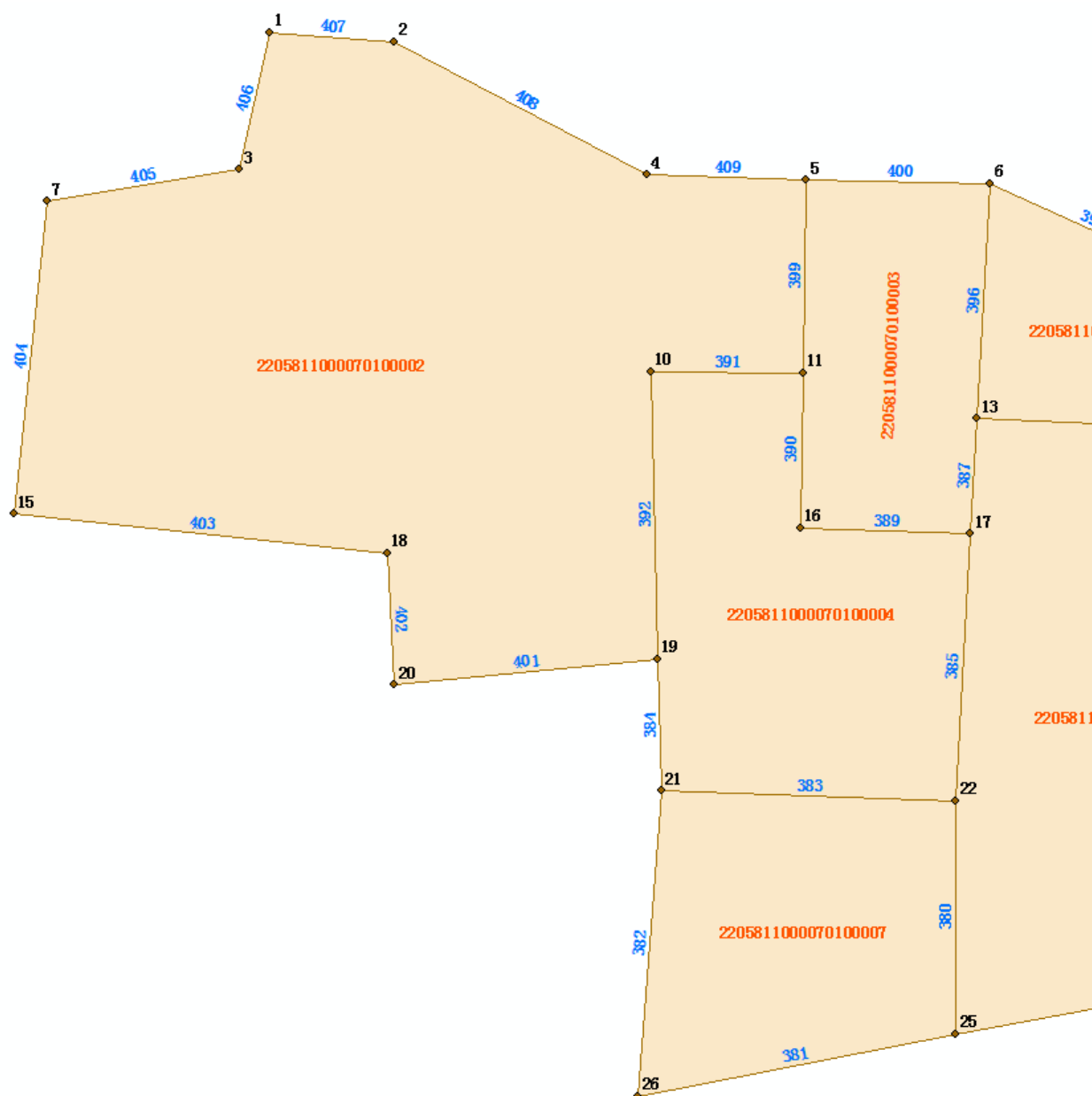
使用 AnchoredSnapper，可以将修正后的 DK 数据连到 Anchor 端口，JZD、JZX 连到请求端口，这样 DK 端口会保持固定不动，将 JZD、JZX 节点移动到 DK 的节点位置。这个转换器也可以使用在 DK 图层的

接边上，假如有两个行政村的 DK 数据，其中一个作为试点村庄，已经发过证书了，另外一个暂时未发证书。如果使用普通的 Snapper 来捕捉，可能两个村的 DK 都会有部分边线发生变化，这样可能会影响到已发证书的面积和数据库面积不一致，如果我们使用 AnchoredSnapper 将已发证的村连到固定端，就能避免这个问题。

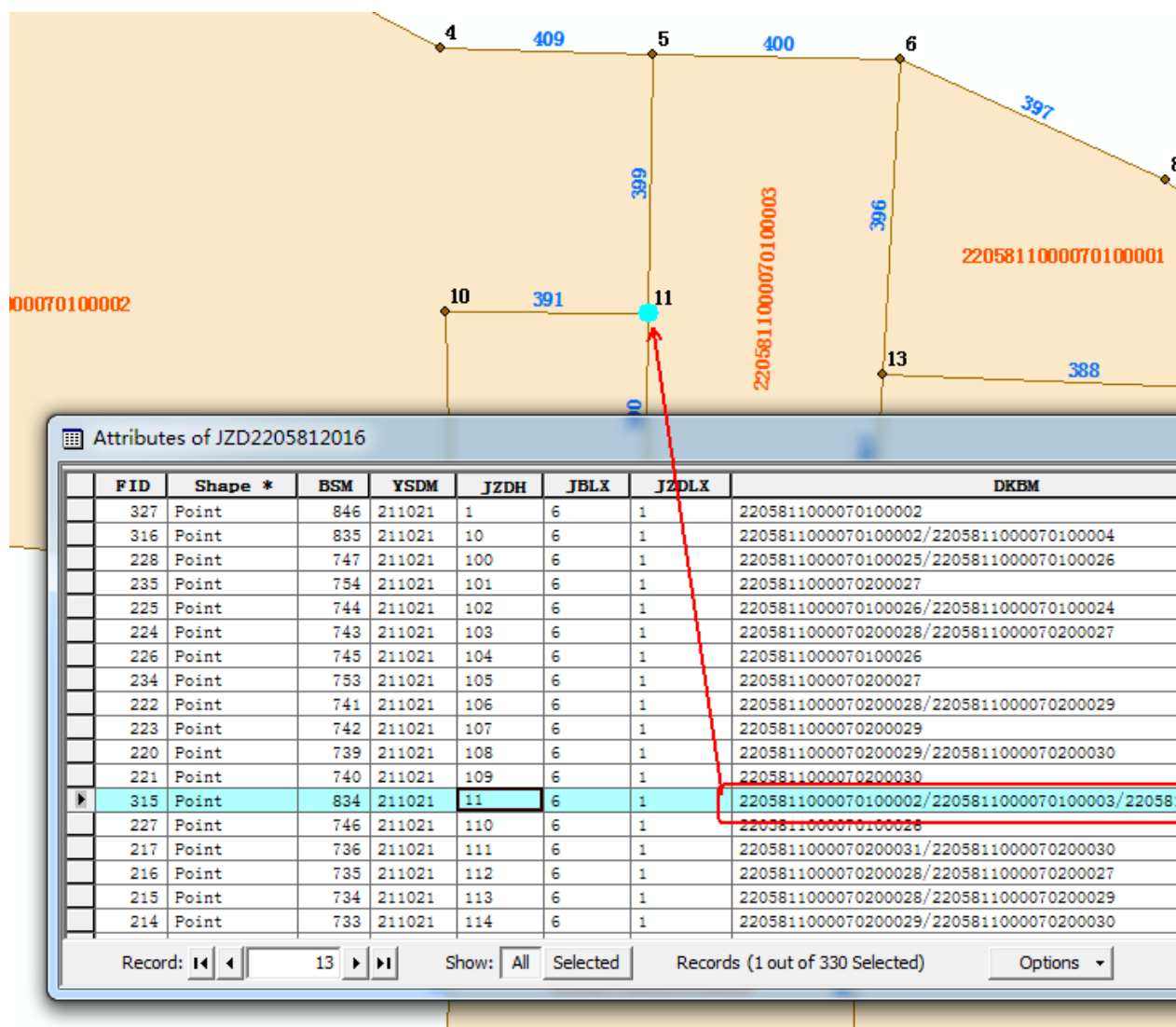
3) 使用 TopologyBuilder 转换器获取 DK 图层的地块编码等信息到 JZX、JZD 图层

DK、JZX、JZD 三个图层关系如果不正确，后期质检可能会出很多矢量错误的问题，例如界址点不在端点上，界址线与地块边线不重合等问题，经过上一步骤处理后，这类问题基本不会存在了，但是矢量图层上有许多信息，需要从点、线、面的关系中获取。JZX 表中，需要得到界址点起止号和相邻地块编号、毗邻权利人名称、指界人等信息，JZD 表需要得到该点涉及到的地块编号信息。具体关系可以参照下图：

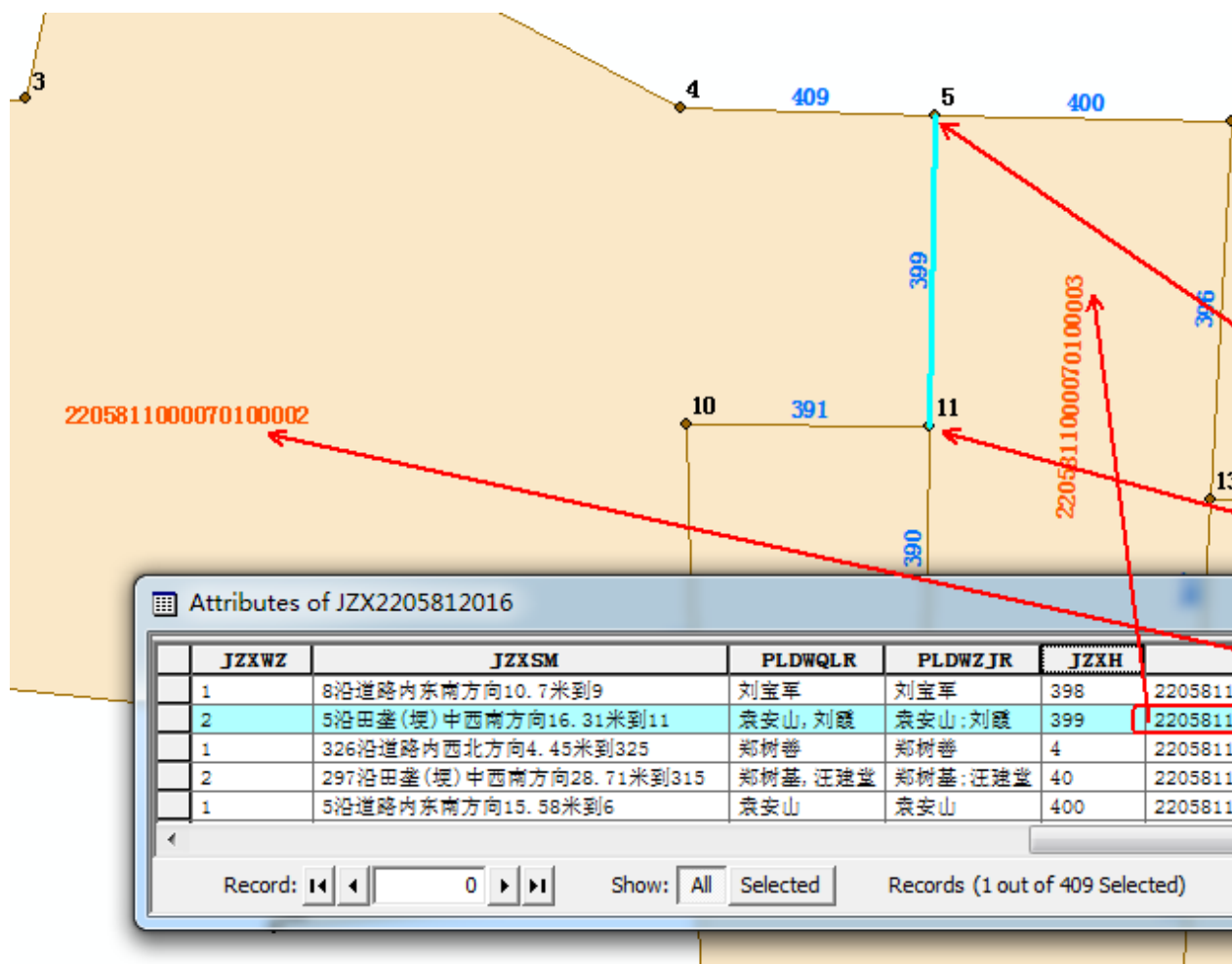
地块、界址点、界址线关系图 1（分布图）



地块、界址点、界址线关系图 2 (界址点表)

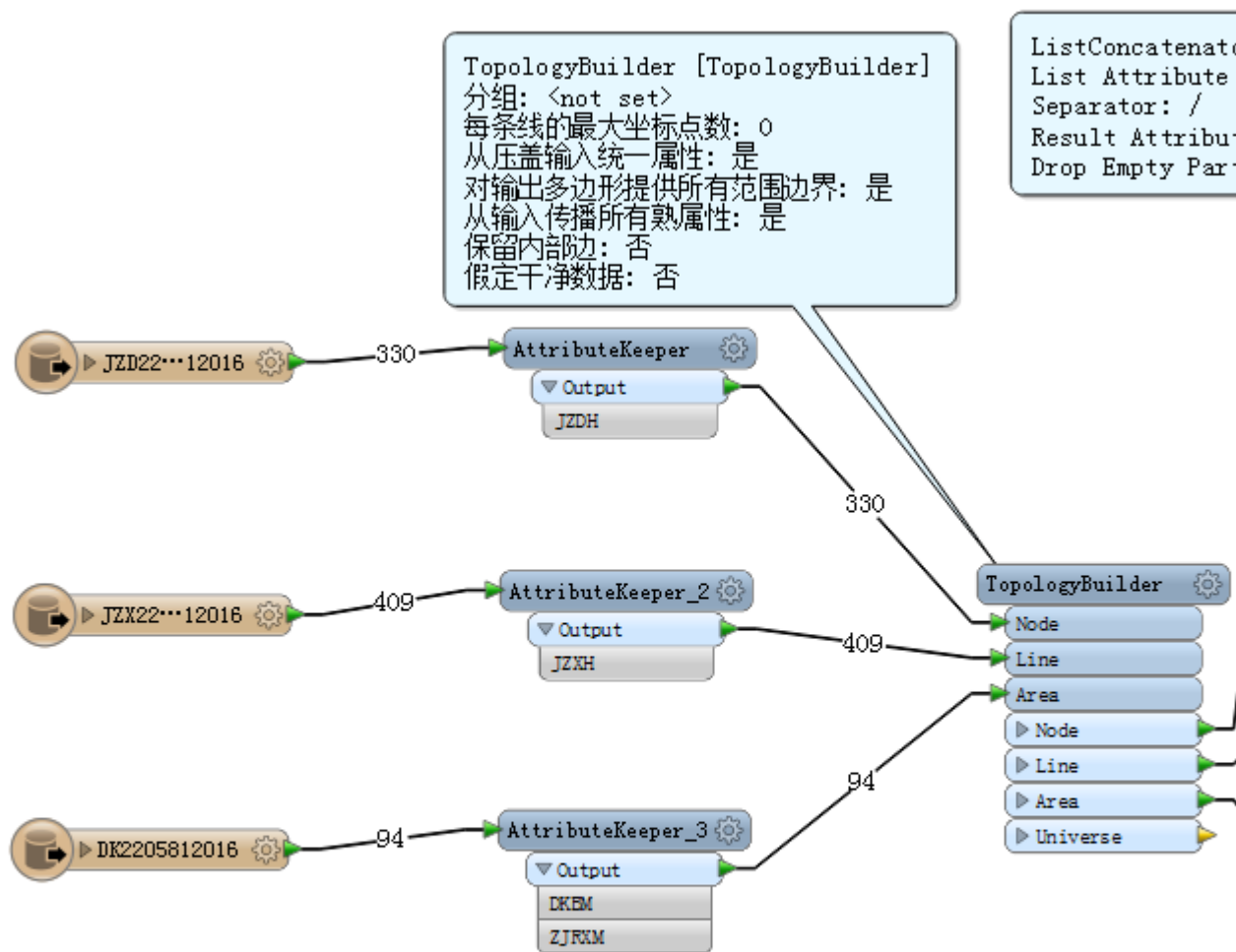


地块、界址点、界址线关系图 3（界址线表）



情况看起来貌似很复杂，交给 FME 只需一个转换器即可获取所有需要的信息，对！他就是 TopologyBuilder。简单模板及输出如下图所示：

TopologyBuilder 示意模板图



界址线表（部分字段）输出：

表视图				
	JZXH	DKBM	起点界址点	终点界址点
392	391	2205811000070100004/2205811000070100002	11	10
393	392	2205811000070100004/2205811000070100002	10	19
394	393	2205811000070100006/2205811000070100001	9	14
395	394	2205811000070100006	9	12
396	395	2205811000070100006	12	23
397	396	2205811000070100001/2205811000070100003	6	13
398	397	2205811000070100001	6	8
399	398	2205811000070100001	8	9
400	399	2205811000070100003/2205811000070100002	5	11
401	400	2205811000070100003	5	6
402	401	2205811000070100002	19	20
403	402	2205811000070100002	20	18
404	403	2205811000070100002	18	15

在 任何列

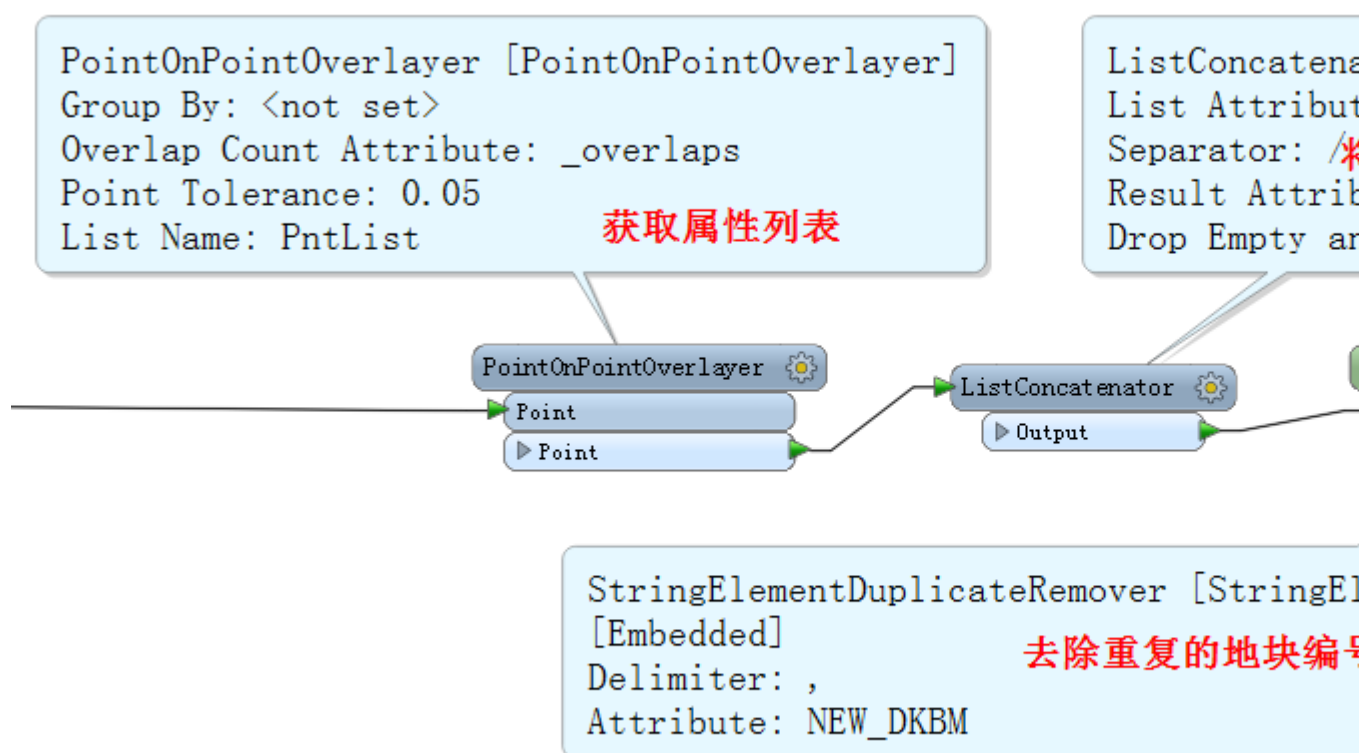
界址点表（部分字段）输出：

	JZDH	DKBM
1	4	2205811000070100002
2	2	2205811000070100002
3	1	2205811000070100002
4	3	2205811000070100002
5	7	2205811000070100002
6	15	2205811000070100002
7	18	2205811000070100002
8	20	2205811000070100002
9	5	2205811000070100002/2205811000070100003
10	8	2205811000070100001
11	6	2205811000070100003/2205811000070100001
12	12	2205811000070100006/2205811000070300060
13	9	2205811000070100001/2205811000070100006

从上表可以看出，JZD、JZX 表所需要的信息使用 TopologyBuilder 一个转换器就可以得到，要特别注意的地方就是，输入数据的拓扑结构一定要正确，否则就不能保证结果正确了，这也是为什么开始需要做接边处理的原因。

4) 接边时属性的处理

在接边的时候，对于 JZD 表，重复的界址点需要删除一个，并将涉及到的地块编码归并到需要保留的界址点上，而对于 JZX 表，也要去除一条重复线并将地块编码归并到保留的界址线上。对于 JZD 表，使用 PointOnPointOverlay 转换器，对于 JZX 表则使用 LineOnLineOverlay 转换器，在叠加转换器的设置上输入列表名称，再使用 ListConcatenator 转换器将含地块编码的字段连接成一个字段，如果可能存在重复（例如多次使用这个模板），最后在输出前，使用 StringElementDuplicateRemover（FME Store 上下载的转换器）对字段进行去重复并排序。具体流程图如下：



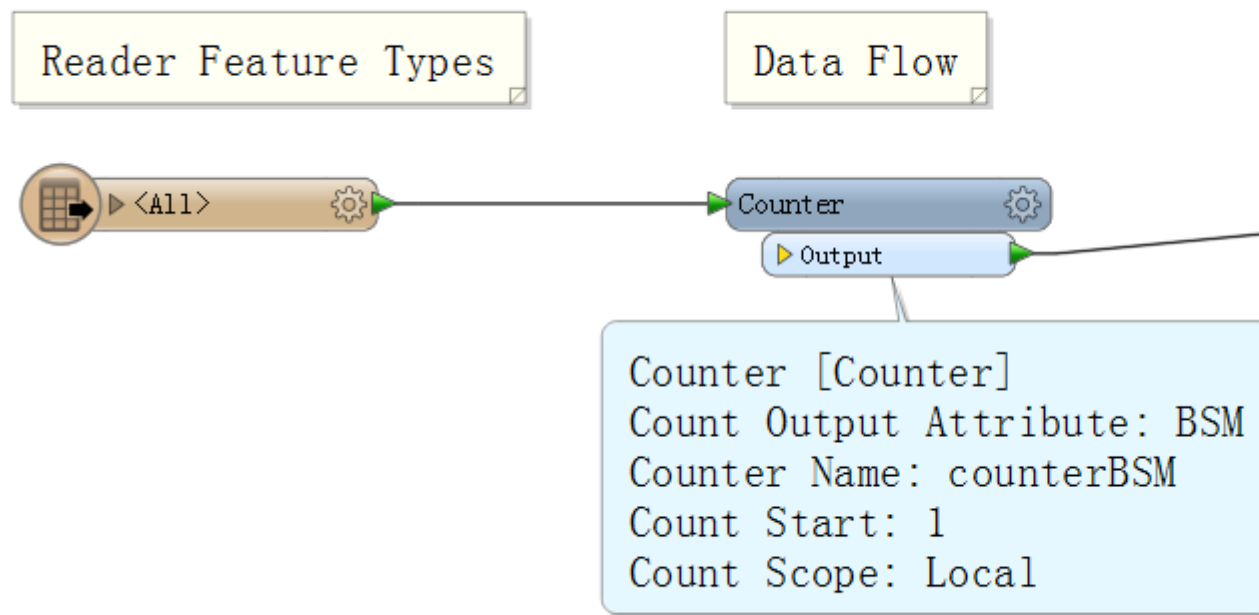
5) 矢量数据 BSM 重编

依据质检规则，全库所有矢量数据 BSM 大于 0 且唯一，也就是说

BSM 的编号 JZD 和 JZX 的两个表之间的 BSM 也不能重复，具体规则是这样描述的：

	属 性 检 查	属性字段值域检查	NY/T 2539-2014 及有关指标修订规定的矢量数据属性字段，其取值必须符合 NY/T 2539-2014 及有关指标修订的规定。对于需要扩展属性值代码的情况，可以进行扩充，但必须能够生成严格满足 NY/T 2539-2014 及有关指标修订要求的标准交换数据。	1、属性字段值域检查要求属性字段取值满足字典表要求，或值大于零。必填字段为空的错误不计入此项检查。 2、除记事类字段，字符串中不能包含空格、回车、制表符构成的字符串，否则作为错误。 3、每个取值超出规定范围的字段作为错误。
		标识码符合性检查	要求各要素层中标识码（BSM）字段值应大于 0，并且 BSM 字段值不得重复，整库唯一。	1、要求 BSM 对于整个确权登记成果数据的各个类型的所有要素在数据库中唯一存在，不得重复。

这种要求对于 FME 来说，实在是小意思了，只需一个 Counter 转换器即可解决问题。其诀窍就是，使用动态的 SHP 读、写模块。具体流程及设置如下：



6) 小结

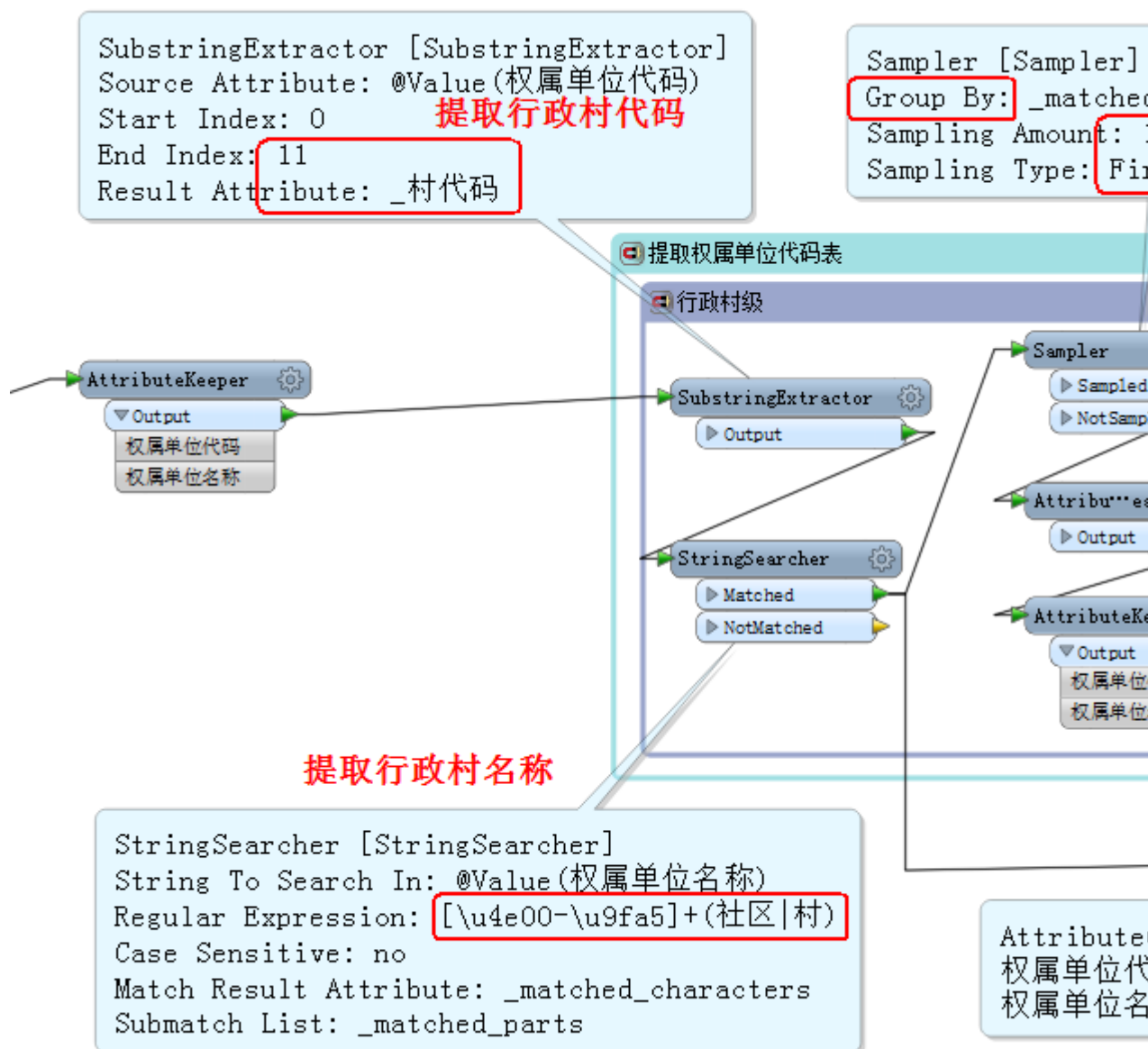
FME 处理图形数据转换器还有很多，远不止提的这几个，真正入库工作中，需要处理的情况很多，要使用多个转换器的组合处理，才能达到需要的效果。另外，由于叠加类转换器均为阻塞式转换器，对内存消耗较大，处理 SHP 数据时，尽量使用 64 位 FME 版本在大内存、性能较好的机器来处理。即使这样，如果全区县的数据并在一起一次性处理，就算内存不爆，耗时也长。对于这种情况，我的方法是分片区处理。这里所指的分片区，并不是指按照行政区划（例如一个镇街）来设置片区，而是依照主要道路、河流等完全隔开的自然分片区。这种片区之间，DK、JZX、JZD 之间不存在需要接边的地方，后期只需要将接边好的数据读取后写到一起就好了，只要不涉及到阻塞式转换器，这种事情哪怕 1G 内存的小机器都能做起来。

三、 使用 FME 处理属性数据

农村土地承包经营权涉及的属性数据分为主要两种，一种是质检目录的权属数据下的以“行政代码+年份”命名的 MDB 数据库，主要是发包方、承包方、地块信息、承包方家庭成员以及权证等信息，这里有些数据是调查时录入的，有些是后期以调查数据为基础统计汇总计算的；另一种是质检目录的汇总表格目录下的相关统计表，这些数据都是以权属单位代码分组，依据权属数据库和矢量图层汇总统计的数据。

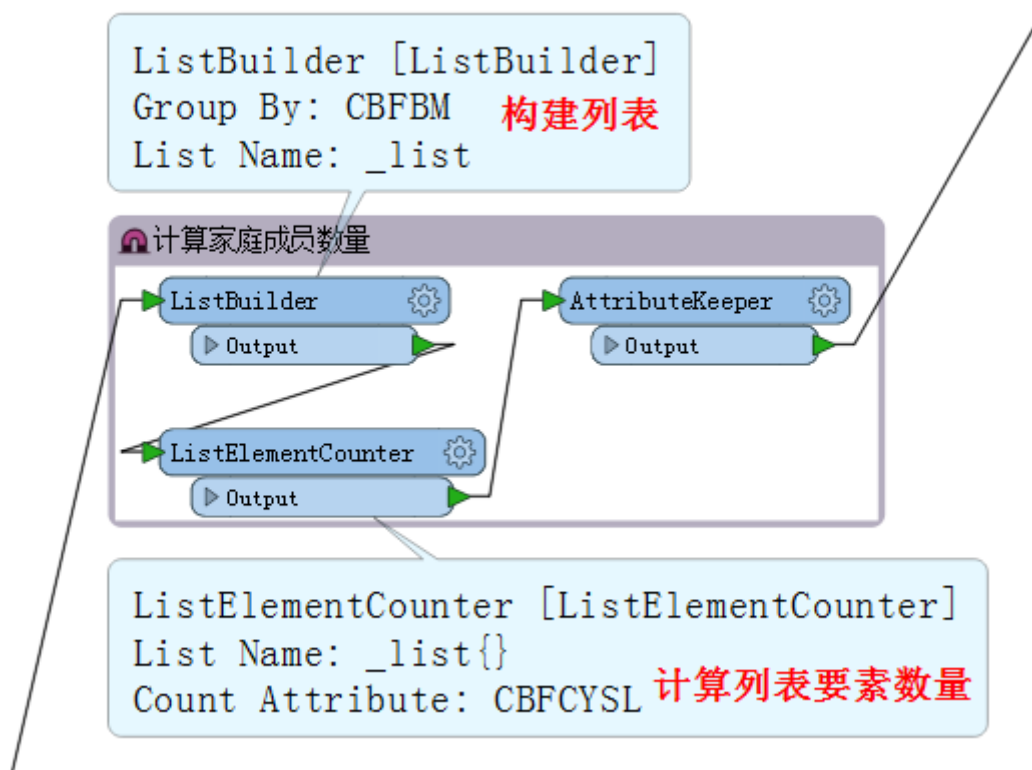
1) 生成权属单位代码表

权属单位代码表实际上是将小组级发包方 (FBF) 里面的 (14 位) 代码加上行政村级 (12 位)、镇街级 (9 位)、区县级 (6 位)，最终权属单位代码不足 14 位的后用 0 补齐。使用 FME 做的话，模板比较简单，涉及的转换器主要是 SubstringExtractor (取字符串---)、StringSearcher (字符串查找---获取相应级别权属单位名称)、Sampler (取样---每组只取一个要素)，模板流程如下：



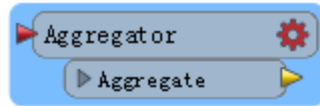
2) 使用 List 方法汇总

属性数据需要汇总的地方比较多，例如合计每个承包方内家庭成员数量、计算承包方地块总数或是统计地块总面积等。具体方法是使用 ListBuilder 建立列表，使用 ListElementCounter 计算列表内要素数量，使用 ListSummer 来求取列表内某个字段的合计，例如计算家庭成员数量部分的模板如下图所示：



3) 使用 Aggregator 方法汇总

List 方法计算适用于需要合计的字段不多的情形，如果需要汇总的字段很多，如果使用 ListSummer 来合计，每个字段需要一个 ListSummer，这种情况下，可以使用 Aggregator 转换器来进行汇总。使用 Aggregator 转换器的属性累积功能，可以实现属性连接、属性求和、求平均数和权重平均值的功能。而且可多选字段，这样就可以实现通过一个转换器，对多个字段进行汇总。



帮助

分隔符

要累积的属性

在这些指定的字段中任何属性将进行累积统计计算。

例如，如果两个输入多边形的属性"salary"分别是30000和50000，那么它们融合多边形的累积计算值为80000。

要求均值的属性

求权重平均值的属性

聚合类型

几何对象名称属性

ID选定

ID属性

父ID属性

子ID列表属性

把属性转换成特点

保存ID

使用方法

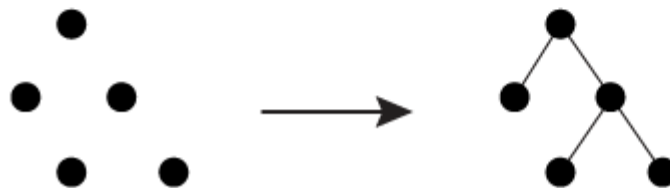
此转换器不是融合共边的面—它只是在给定条件下创建一个简单的要素集合。若需要合并相邻面为一个面要素。

举例

几何表征



数据结构表征



以发包方编码为分组条件,合计地块总数和总面积的转换器设

置如下：

Aggregator Parameters

Transformer

Transformer Name: Aggregator_2

Group By: FBFBM

Input is Ordered by Group: No

Parameters

Mode: Attributes Only

Count Attribute:

Attribute Accumulation **以FBFBM分组，同时汇总地块总数和总面积**

Keep Input Attributes: Yes

List Name:

Attributes to Concatenate: No items selected.

Separator Character: ,

Attributes to Sum: 承包地块总面积 承包地块总数

Attributes to Average: No items selected.

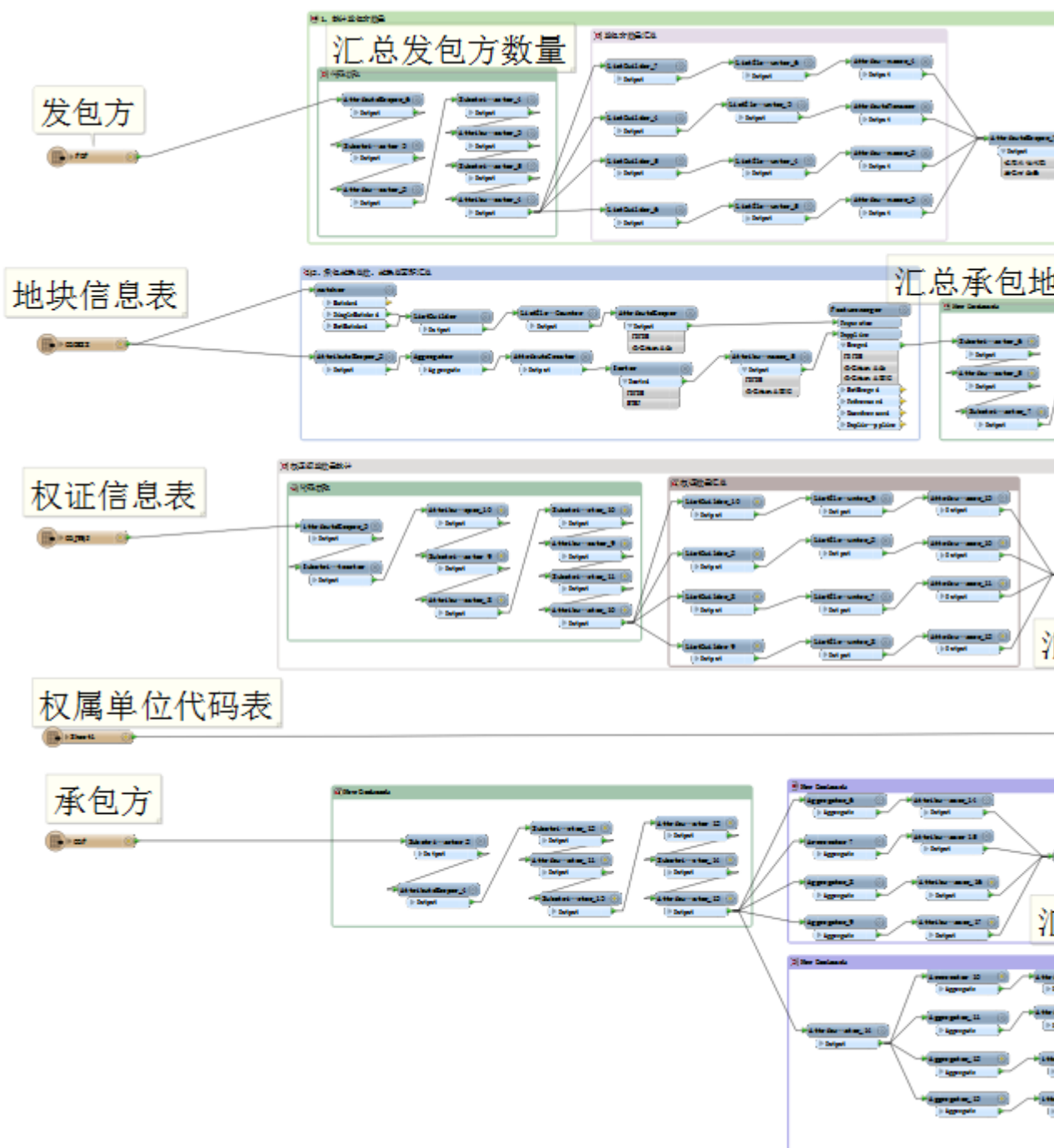
Attributes to Average, Weighted by Area: No items selected.

Assemble One Level

Assemble Hierarchy

Help Defaults OK Cancel

将各项需要汇总的数据，分别用权属单位代码分组汇总，最后使用 FeatureMerger 合并起来，即可形成多项数据的汇总表。统计汇总信息的模板流程如下：



4) 小结

和大多数项目一样，收官验收阶段，需要汇总统计的表格会很多。使用 List 或是 Aggregator 统计，难度都不大，使用哪个转换器主要看个人习惯。这两个转换器都是阻塞式的转换器，对于大数据的处理，个人认为使用 Aggregator 的内存需求比 List 的要小一些，可能是因

为 ListBuilder 内存需求更大吧。我对于数量较少的一般使用 List，而大数据量的则使用 Aggregator 来进行汇总。在统计记录数时，List 可以直接使用 ListElementCounter 来计算列表内要素数量，Aggregator 则不行，但是可以在汇总前，新建一个要素数量的字段，赋值为 1，汇总后即可得到要素数量。另外 Aggregator 也可以设置保存列表，灵活性更好一些。

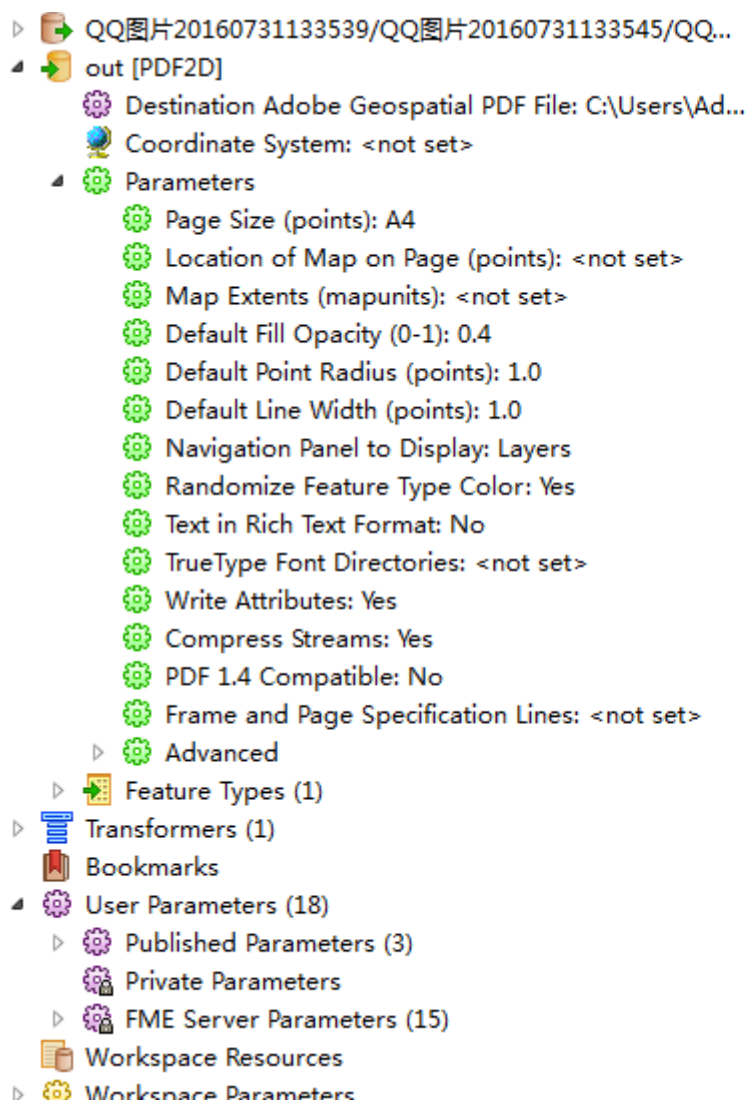
四、 使用 FME 处理图件数据

图件数据包含两类，一类是 JPG 格式，另一类是 PDF 格式。JPG 格式一般用于单页的情况，如果是多页数据的话，可使用 PDF 文件作为图件存放格式。质检标准格式的“图件”目录下主要是存放地块示意图，另外还有一种在“其他资料\权属来源资料”下存放的，是土地权属调查时收集的承包方身份、户籍信息以及土地承包合同等资料的影印件副本。由于原国标要求的上报图件数据均为 JPG 格式，PDF 格式是后期改版后才增加的，所以有不少数据需要从多张 JPG 格式转换到 PDF 的多页格式。入户权属调查时，普遍做法是使用相机拍照方式将承包方身份、户籍、权属来源证明材料采集，以便作为权属来源资料进行归档，这就需要按照一定顺序对资料进行排版后保存为 PDF 文件，例如将第一页放置身份证的正反页，第二页开始放置户口薄，然后接着是权属来源资料，最后放置其他证明材料。有些地方的地块示意图的格式为以发包方为单位存放 Word 的 DOC 格式，每个文件包含若干承包方的地块示意图，这就需要将 DOC 格式进行拆分成以承包方为单位的多页 PDF 格式。矢量图件转成 PDF 或是 JPG 格式，这种就

不在此介绍了，这里主要介绍一下使用 FME 将 JPG 格式转变成多页 PDF 并进行排版以及使用 FME 对 PDF 文件进行合并、分割的方法。

1) 多个 JPG 转成多页 PDF

单纯地将多个 JPG 文件直接存为多页 PDF，还是比较简单的，主要是设置一下 PDF 的输出参数即可，合并读取多个 JPG 文件进来后，使用 Counter 计数转换器对每张 JPG 文件编号，编号字段名称设置为 pdf_page_number，从 1 开始计数，然后在 PDF 写模块上暴露格式属性即可。如果其他不做设置，默认生成的 PDF 页面上最大尺寸 JPG 占页面尺寸的 90%，简易模板如下：



如果需要将图形填充整个 PDF 页面，需要设置 PDF 写模块的 Location of Map on Page 参数，对于纵向的 A4 纸张，该值为“0 0 595 842”，这个值分辨率是 72 像素/英寸时，A4 纸的尺寸的图像的像素是 595×842。如果输入的 JPG 为横向，将图形旋转 90 度即可。

2) 对 JPG 进行排版后转成多页 PDF

如果是简单的 JPG 转 PDF，也有现成的软件可以完成。但是如果是需要排版、还要定义顺序、甚至实现批量处理，如果不使用 FME，那就只能通过编程来完成了。对于权属调查收集的资料，只要将图片纠偏、裁剪、保存为命名有规律的 JPG 文件后，剩下的事情就可以交给 FME 了。下面以一个实例来介绍一下具体流程：

该项目现有数据及最终要求如下：

- a) 已经裁剪好的 JPG 图片格式若干，身份证文件命名规则为 CBFBM_CBFMC_身份证(0-n).jpg, 户口簿命名规则为 CBFBM_CBFMC_户口簿(0-n).jpg, 权证命名规则为 CBFBM_CBFMC_权证_权证名称(0-n).jpg, 其他证明材料命名规则为 CBFBM_CBFMC_其他材料_证明名称(0-n).jpg, 其中身份证(原承包方、现承包方, 正反面)最多四张图片、户口簿(张数不定)为必须提供。
- b) 输出数据为多页 PDF
- c) 第一页为身份证，正反面放置，尺寸固定为 85.6mm*54.0mm
- d) 第二页开始为户口簿，一页最多放置两张，尺寸为

143mm*105mm

e) 户口薄后放置权证图片，横向、可一页两张

f) 最后放置其他证明文件，纵向、一页一张

从要求可以看出，可以固定页码的只有第一页和第二页，其他的都是依据最大号码顺序编号，模板需要解决三个问题：一是输入顺序问题，需要将身份证、户口薄、权证、其他证明材料，按照顺序输入进来再写入 PDF 文件；二是 PDF 页码编排问题，对于不固定的页码，需要从上一材料的最大号来顺序编号，如何获取上一材料的最大编号；三是 PDF 版面排放的问题，如何判断一页放两张或是一张，横向或是纵向。

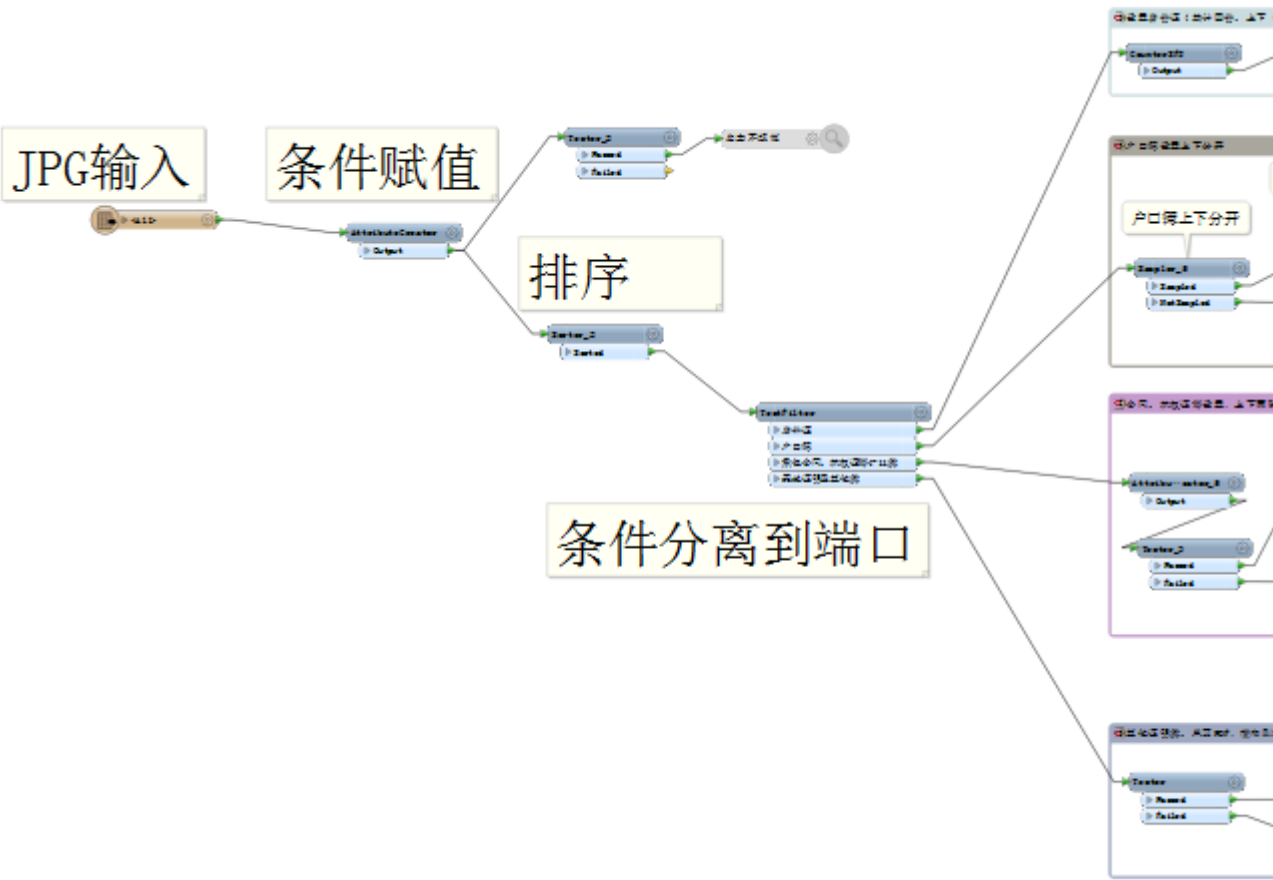
对于输入顺序，直接读取肯定是不行的，那就需要将数据先进行归类并赋值，这里可以使用 AttributeCreator 转换器的条件赋值功能，将类型数字化，例如身份证类赋值 1，户口薄类赋值 2，权证类赋值 3，其他材料赋值 4，然后使用编码、文件名使用 Sorter 转换器来排序，设置编码为第一排序规则。这样排序后的要素就是按照要求来的。将排序后的要素使用 TestFilter 转换器进行分类，经过这个转换器直接可以分成四个端口，后期针对四种情况分别处理即可。

页码编排和最大页码传递的问题，身份证页码固定是 1，户口薄页码固定从 2 开始，后续的问题是将户口薄最大页码传递到权证类，权证类最大编码传递到其他证明类。我采用的方法是变量发布、提取的组合转换器，即 VariableSetter 和 VariableRetriever 转换器的组合。在每编一个户口薄页码后，就将该页码使用 VariableSette 转

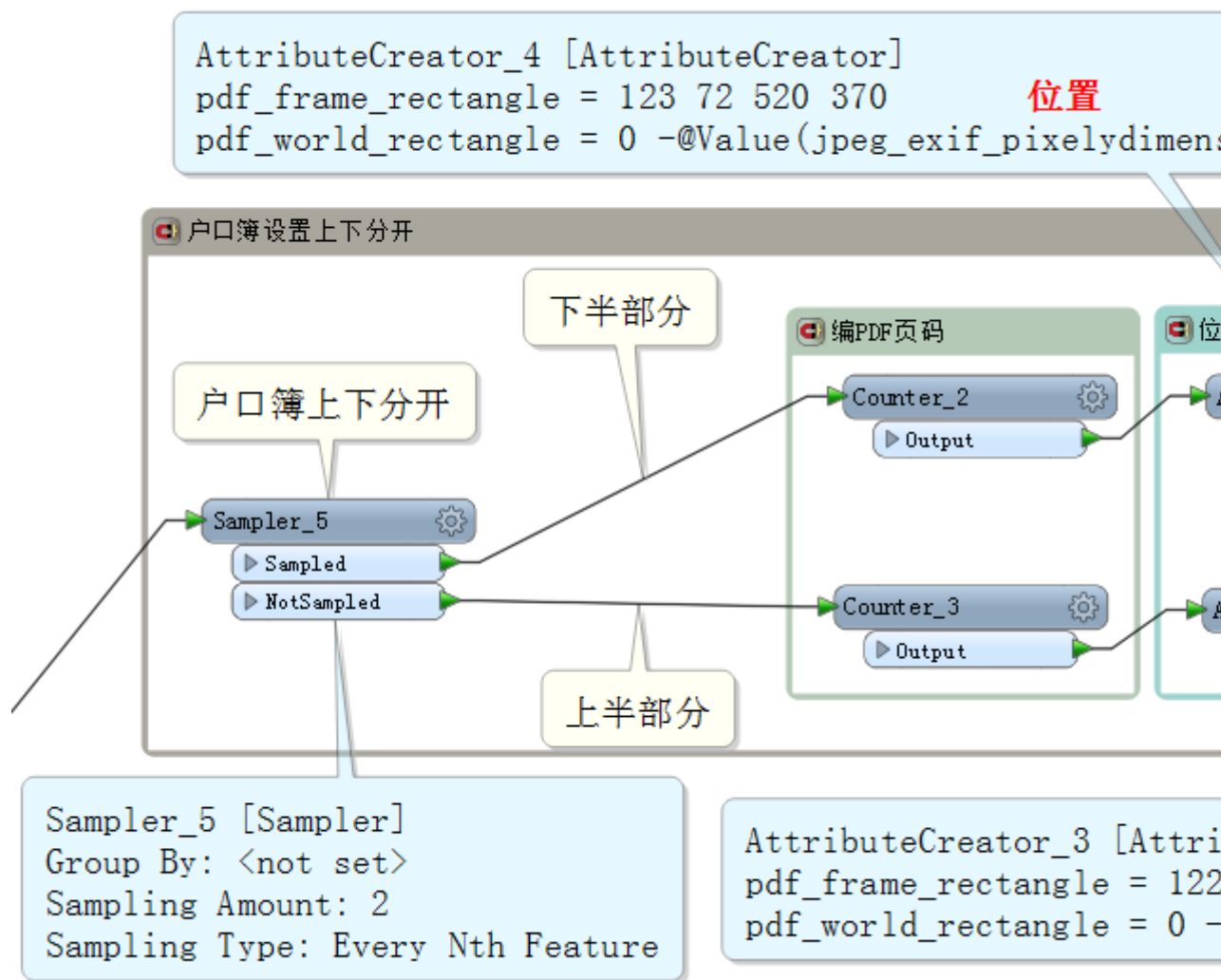
换器发布到变量，这样当模板顺序运行到权证类时，编号前使用 VariableRetriever 把值取回来就可以编号了，同理，其他证明的起始编号也可以由权证类的最大号传递过来。

关于排版时，一页排两栏的如何处理，我的方法是使用 Sampler 取样转换器，设置逢 2 取 1，即可分离成两个端口，取样的那个端口设置输出到 PDF 页面上半部分，丢弃的那部分输出到 PDF 页面的下半部分。关于输出到 PDF 页面后尺寸大小的设置，经过查询写模块参数得知，需要设置 pdf_frame_rectangle、pdf_world_rectangle 两个参数。

模板流程如下：



户口薄部分截图如下：



生成结果如下：



3) 使用 FME 拆分、合并 PDF 文件

前面介绍过，有些地方的地块示意图是以 Word 的 DOC 文件来保存的，而且是一个发包方作为一个文件来存放的。Word 文件可以方

便地转换为 PDF 文件，也可以实现选择多个页面另存为 PDF,如果需
要将每户涉及到的页面，逐户导出为单独的 PDF，需要编程才能解决。

熟悉 FME 的朋友都知道，FME 对 PDF 的支持，仅限于写出，读取
到目前是不支持的。FME 可以写出 2D 和 3D 的 PDF 文件，那么如何使
用 FME 来拆分或者合并 PDF 文件呢。这里就需要使用第三方软件。FME
对其他软件的支持有许多种，例如 Python、R 语言等，这些是属于嵌
入式的，使用类似 PythonCaller、PythonCreator 之类的转换器，来
实现需要的功能。我这里所指的第三方软件，并不是嵌入式的开发语
言，是指具有命令行运行模式的可执行文件，通过 FME 求取需分割的
PDF 页码范围及文件名称，编写相应命令行语句，使用 SystemCaller
转换器来执行该语句，间接地达到分割或者合并 PDF 文件的效果。

这个软件就是 PDF Split/PDF Merge，该软件既可以在 Windows
环境下以图形界面执行，也可以使用命令行传递参数来执行。该软件
的 下 载 地 址 为
<http://www.adultpdf.com/products/pdfsplitlemerge/index.html>，
具体参数设置方法，该软件使用自定义页码或范围拆分 PDF 文档部分
的帮助说明如下：

**Split PDF documents by page numbers or custom page
ranges.**

<input>

input file path, eg. c:\abc\input.pdf

<range>

indicate extract pages of ranges from a PDF file, e.g.

1,3,5-10;2,4,6;8-10 ,

please refer to split parameters table above.

[output base]

output file base name, eg. c:\aaa\output

example:

```
pdfspme_cmd.exe -spr "c:\abc\input.pdf" "1,3;5-10;2,4,6  
,8-10" "c:\aaa\output"
```

if input.pdf have 10 pages, then result is:

c:\aaa\output01.pdf file content include page1 and
page 3.

c:\aaa\output02.pdf file content include 5-10 of the
page.

c:\aaa\output03.pdf file content include
page2,4,5,8,9,10.

```
pdfspme_cmd.exe -spr "c:\abc\input.pdf" "1,3;8-" "c:\aaa\o  
utput"
```

if input.pdf have 10 pages, then result is:

c:\aaa\output01.pdf file content include page1 and page

3.

c:\aaa\output02.pdf file content include page 8 - the end
page.

从所需要的参数我们可以看出，只要知道每个需要拆分的起止页码，就能将 PDF 对应的页码范围提取并另存到一个 PDF 文件。对于地块示意图文件来说（前提是文件依据排序顺序来存放，例如按承包方编码顺序），获取农户的页码，只要知道每户地块总数，幅面的最大地块数，即可求出每户会有几页。然后依据顺序即可算出该文件涉及多少户，共有多少页，每户的起止页码范围。知道页码范围后，就可以使用命令行来执行拆分了。我所经手的项目，使用的方法是编程直接由 Word 获取页面内承包方代码后，直接转存为 PDF，而且这个模板也不是太复杂，这里就不详细讲解了。需要注意的一点是，我在测试时，使用 SystemCaller 直接运行命令行时，会提示参数过长的错误，这种情况的解决办法是，先使用 AttributeFileWriter 将命令行写出成 BAT 文件，再用 SystemCaller 来调用就可以了。

4) 小结

这段主要介绍了 FME 和 PDF 的一些操作，PDF 文件在质检时，不用检查分辨率，而 JPG 文件需要 DPI 达到 300 以上，否则会有分辨率不够的警告信息，具体采用什么还是要依据实际情况来判断。使用 SystemCaller 来调用第三方软件，运行命令行可以用来处理一些 FME 不能直接做的事情。

五、 使用 FME 实现自动入库

我之所以选择 FME 来做数据入库，数据支持格式多、模板调试方便是一方面，最主要的原因是只要你的模板调试好了，就可以很方便地部署模板批量运行。有些软件，某些功能只能单个处理，无法批量生产，这对作业效率可是致命的。FME 模板的批量运行，有三种方法可以，一是使用模板嵌套，在模板内调用 WorkspaceRunner 转换器运行另一个 FMW 文件，传递参数来运行，该方法的优点是操作简单、界面直观、还能多线程运行，缺点是无法运行加密的 FMW 和 “.fme” 映射文件，有时候还不太稳定，我使用过程中还遇到过死机的情况；二是使用 FME 软件批量部署 (Batch Deploy...) 功能，该方法是依据用户选择，生成 TCL 控制文件，从而达到批处理，缺点是不够灵活，而且 TCL 用得太多，不熟悉；第三种就是使用 BAT 文件，控制行命令传递参数来运行，这种方法部署方便，还能够使用 “.fme” 映射文件运行，通过使用 FOR 循环和 CALL 语句，能够快速、方便地部署一个高效的批量处理运行机制，甚至可以达到无人值守的级别，缺点也有，就是因为他功能太多了，批处理命令到现在还只学会了一小部分，但是就是这一小部分，也能实现好多功能。下面开始实例介绍：

1) 批量循环运行单个模板

如果不循环，BAT 实际意义并不大，以刚才 JPG 转多页 PDF 为例，该模板共有两个参数，读模块是读取承包方 JPG 目录下所有的 JPG 文件，写模块为单个 PDF 文件，使用 BAT 来运行，首先需要获取所有承包方的 JPG 存放目录，然后循环运行该模板，将输入文件和输出文件参数传递到模板即可。该模板的运行使用的命令行是类似这样的：

```
fme.exe d:\jpeg2pdf2d_V2015_b15515+.fmw --S_JPEG  
"d:\test\cbf\32010100100101_张三\*.jpg" --D_PDF2D "  
d:\test\32010100100101_张三.pdf"
```

我们可以在 d:\test\cbf 目录下建立一个 BAT 文件，首先要获取 \cbf\ 目录下所有的子目录，然后将子目录后加上 “*.jpg”，传递给参数 --S_JPEG，将子目录后加上 “.pdf”，将值传递给参数 --D_PDF2D。具体需要使用一个 DIR 的 DOS 命令和 FOR 循环语句。获取所有子目录的 DOS 命令是 “DIR /A:D /B”，把 DOS 命令加入 FOR 循环后，BAT 内容如下：

```
@echo off&color 0a&title 批量 JPG 转 PDF  
  
echo                                     设置说明  
  
echo  
  
-----  
  
echo 注意:用记事本打开本文件修改所有 FME 安装路径为你的  
本机 FME2015 安装路径  
  
echo      当前运行模板为 jpeg2pdf.fmw  
  
echo  
  
-----  
  
set fmeapp="D:\apps\FME15515\fme.exe"  
  
for /f "delims=" %%i in ('dir /a:d /b') do (
```

```

echo 现在正在处理%%i.....

%fmeapp% ".\jpeg2pdf.fmw"^ --S_JPEG ".\%%i\*.jpg"^
--D_PDF2D ".\%%i\%%i.pdf"
)

echo JPG 转换 PDF 已完成！数据存放规则为\DIR\DIR.PDF

pause

BAT 文件的循环主体是 for /f "delims=" %%i in ('dir /a:d /b')
do (), do() 里面的就是需要循环运行的命令行，其中输入输出参数
均用变量代替。

```

目录结构及运行界面如下：

名称	修改日期	类型	大小
 TEST001	2017/3/5 19:55	文件夹	
 TEST002	2017/3/5 19:55	文件夹	
 TEST003	2017/3/5 19:55	文件夹	
 jpeg2pdf.fmw	2017/1/23 1:09	FMW 文件	134 KB
 jpeg2pdf.log	2017/3/5 19:57	文本文档	34 KB
 按目录批量生成PDF.bat	2017/3/5 19:56	Windows 批处理...	1 KB

```
管理员: 批量JPG转PDF

设置说明

-----
注意:用记事本打开本文件修改所有FME安装路径为你的本机FME2015安装路径
      当前运行模板为jpeg2pdf.fmw
-----

现在正在处理TEST001.....
      FME 2015.1.1.0 <20150715 - Build 15515 - WIN32>
      FME Desktop ESRI Edition <floating>
      Permanent License.
      Machine host name is: Administrator
      Operating System: Microsoft Windows 7 64-bit Service Pack 1 <Build 7601>
      Copyright (c) 1994 - 2015, Safe Software Inc.
      Safe Software Inc.

Reading...
Emptying factory pipeline....
.....Translation was SUCCESSFUL
现在正在处理TEST002.....
      FME 2015.1.1.0 <20150715 - Build 15515 - WIN32>
      FME Desktop ESRI Edition <floating>
      Permanent License.
      Machine host name is: Administrator
      Operating System: Microsoft Windows 7 64-bit Service Pack 1 <Build 7601>
      Copyright (c) 1994 - 2015, Safe Software Inc.
      Safe Software Inc.

Reading...
Emptying factory pipeline....
.....Translation was SUCCESSFUL
现在正在处理TEST003.....
      FME 2015.1.1.0 <20150715 - Build 15515 - WIN32>
      FME Desktop ESRI Edition <floating>
      Permanent License.
      Machine host name is: Administrator
      Operating System: Microsoft Windows 7 64-bit Service Pack 1 <Build 7601>
      Copyright (c) 1994 - 2015, Safe Software Inc.
      Safe Software Inc.

Reading...
Emptying factory pipeline....
.....Translation was SUCCESSFUL
JPG转换PDF已完成! 数据存放规则为\DIR\DIR.PDF
请按任意键继续. . .
```

2) 批量顺序运行多个模板

一项工作，往往不是一个模板能够解决的，需要进行多个步骤才能达到要求，有时候需要将等 A 模板运行完成后，调用 A 模板的结

果再来执行 B 模板。这样就需要配置好目录结构和文件命名，以便批处理文件来识别和读取。前段时间做的一个数据转换就是如此，该项目提供了两项数据，一是地方标准的 EXCEL 格式的属性文件，含发包方、承包方、家庭成员以及共用地块分摊表格，二是地方标准的 SHP 格式的 DK 数据。其中 EXCEL 表格是以发包方为单位，一个发包方一个文件，SHP 数据以行政村为单位，一个村大约包含 30 个发包方。需求就是将 EXCEL 格式以及 SHP 格式的文件转换到农经权标准质检目录下国标格式。

由于 EXCEL 文件个数很多，相关数据要统计的话很不方便，我采用的方法是先将所有的 EXCEL 表格导入到一个中间 MDB 数据库，然后利用中间数据库进行汇总后写入国标数据库，SHP 数据也需要接边处理后，添加 JZD、JZX 相关信息再写成国标 SHP 格式。这里需要使用三个模板，即 EXCEL2MDB，SHP2GBSHP，MDB2GBMDB。其中 EXCEL2MDB 批量运行时，需要使用 FOR 循环来处理。而且输出后文件目录存放和命名需要有规则。各个 BAT 文件配制如下：

名称	大小	类型	修改日期	属性
 361101010101		文件夹	2017-1-6 17:40	
 EXCEL表格		文件夹	2017-1-6 22:32	
 FME模板		文件夹	2017-2-8 10:49	
 MDB国标库		文件夹	2017-2-8 10:50	
 MDB中间库		文件夹	2017-2-8 10:49	
 SHP国标		文件夹	2017-2-8 10:50	
 SHP原始		文件夹	2017-1-6 17:40	
 检查结果		文件夹	2017-2-8 10:49	
 1、EXCEL导入MDB中间库.bat	1 KB	Windows 批处理文件	2017-3-5 20:50	A
 2、检查MDB中间库.bat	1 KB	Windows 批处理文件	2017-1-7 0:15	A
 3、MDB中间库转国标MDB数...	1 KB	Windows 批处理文件	2017-1-7 0:07	A
 4、原始SHP转国标SHP点线...	1 KB	Windows 批处理文件	2017-1-7 0:15	A
 all.bat	1 KB	Windows 批处理文件	2017-1-6 22:26	A

```
1、EXCEL导入MDB中间库.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
@echo off&color 0a&title EXCEL转MDB临时库
echo                                     设置说明
echo
echo 注意:用记事本打开本文件修改所有FME安装路径为你的本机FME2015安装路径
echo 当前运行模板为xlsxr2mdb_ado.fme
echo

set fmeapp="C:\apps\FME2015.1.1\fme.exe"
del .\MDB中间库\EXCEL合并数据库.mdb

for /f "delims=" %%i in ('dir /a-d /b /s ".\EXCEL表格\*.xls"') do (
%fmeapp% ".\FME模板\xlsxr2mdb_ado.fme" ^ --SourceDataset_XLSXR "%%i" ^ --DestDat
)

3、MDB中间库转国标MDB数据库.BAT - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
@echo off&color 0a&title MDB中间库转国标数据库
echo                                     设置说明
echo
echo 注意:用记事本打开本文件修改所有FME安装路径为你的本机FME2015安装路径
echo 当前运行模板为MbdToGB.fme
echo

set fmeapp="C:\apps\FME2015.1.1\fme.exe"

%fmeapp% ".\FME模板\MbdToGB.fme" --S_MDB_ADO ".\MDB中间库\EXCEL合并数据库.mdb"
--S_SHAPE ".\SHP原始\*DK.shp" --OUT_MDB_FILE ".\FME模板\国标权属数据库.mdb" --D

echo 第三步检查MDB中间库转国标数据库已完成!数据存放在\MDB国标库\文件夹中

4、原始SHP转国标SHP点线面.BAT - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
@echo off&color 0a&title 原始SHP处理后转国标SHP
echo                                     设置说明
echo
echo 注意:用记事本打开本文件修改所有FME安装路径为你的本机FME2015安装路径
echo 当前运行模板为SHPToGB.fme
echo

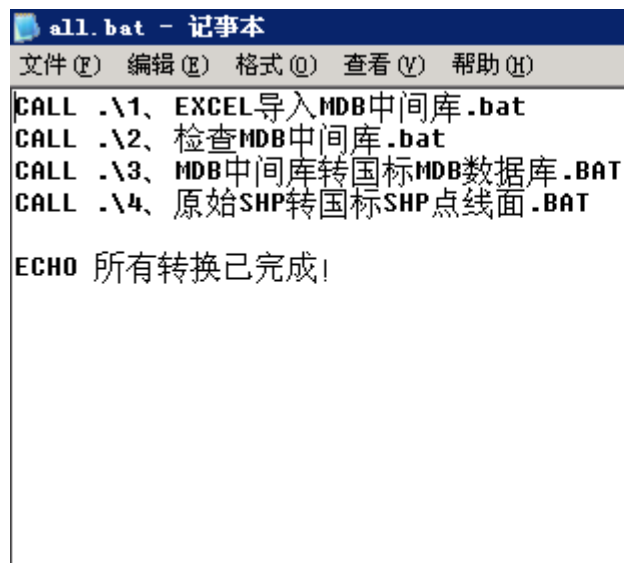
set fmeapp="C:\apps\FME2015.1.1\fme.exe"

%fmeapp% ".\FME模板\SHPToGB.fme" --S_SHAPE_DK ".\SHP原始\*DK.shp" --D_SHAPE ".\

echo 第四步原始SHP处理后转国标SHP已完成!数据存放在\SHP国标\文件夹中
```

这些是单个可执行的 BAT 文件，如果需要单独执行哪个就点击哪

个，如果是顺序运行，则可以点击 ALL.BAT，该 BAT 文件实际就是使用 CALL 语句顺序调用各个 BAT 文件，配置如下：

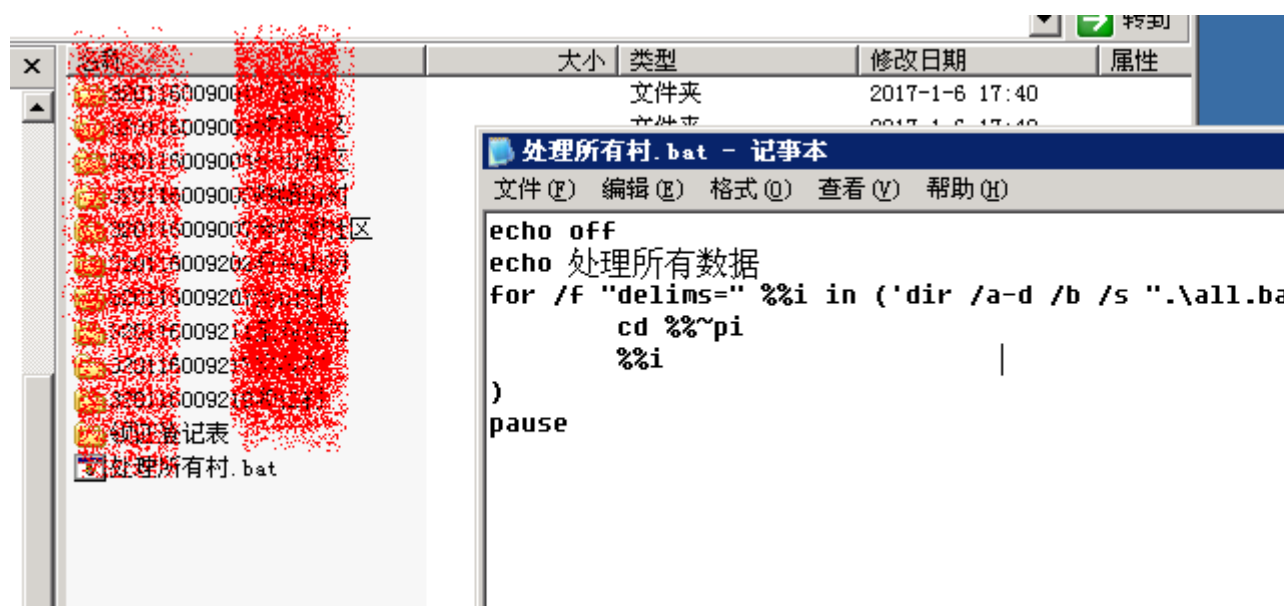


```
all.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

CALL .\1、EXCEL导入MDB中间库.bat
CALL .\2、检查MDB中间库.bat
CALL .\3、MDB中间库转国标MDB数据库.BAT
CALL .\4、原始SHP转国标SHP点线面.BAT

ECHO 所有转换已完成!
```

这些 BAT 文件是存放在行政村级别目录下的，运行一次 ALL.BAT 则可以将该行政村下 30 多个发包方的属性数据转入临时 MDB 库，然后汇总统计后生成国标 MDB 数据库，SHP 数据也是如此，处理后将国标 SHP 输出到该行政村固定目录下。那么能否更智能一些呢，按行政村一个个顺序执行下去呢。当然可以了，只需要在上一级目录下建立另一个 BAT 文件即可，该 BAT 文件内容如下：



```
处理所有村.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

echo off
echo 处理所有数据
for /f "delims=" %%i in ('dir /a-d /b /s ".\all.ba
    cd %%~pi
    %%i
)
pause
```

这个批处理文件查找目录下所有的 ALL.BAT，然后进入该目录后执行 ALL.BAT 文件，为什么需要进入目录再执行呢，因为在写 BAT 时，用的是相对路径，不是绝对路径，如果不进入 ALL.BAT 所在的目录，会出现找不到文件的情况。使用这种方法，可以一次处理上若干个行政村。可以在机器闲置的时候，开始部署运行，BAT 会按照设定的模式，顺序执行，直至全部执行完毕。经测试一个乡镇大约 10 个行政村的数据，全套流程走下来，耗时约一个小时左右。

3) 小结

使用 BAT 方法提高效率，结果是显而易见的，但是在使用上也要注意，那就是 log 文件，你无人值守时，模板运行是否成功，中间是否含有警告？光从 BAT 是看不出来的，因为单个模板执行出错，不会影响后续模板的执行，所以需要分析日志文件。对于非循环的批处理，单个模板运行的情况下，一般 log 文件即为“模板名称.log”，而对于循环执行的模板，则需要将 LOG 参数发布出来，供后期分析使用。BAT 运行方式内存占用比传统方式内存节省，毕竟少开了一个 Windows 窗口，关于日志分析的方法，可以参考 FME 模板兴趣班作业。

六、 使用 FME 实现质量检查

质检软件是专业做检查的，还需要 FME 做检查模板么？由于侧重点不同，质检的查错模式和报错方法并不一定适合每种结果。而且有一些错误，在并库之前修改是最好的，比如界址点距离过近，在没有生成 JZD、JZX 之前，查出来只需要修改 DK 就可以了，如果后期质检

再查出来，点、线、面都需要修改。类似于质检软件报出的错误，XXXXXXXX 号身份证存在重复值，改错的时候，需要先查找，再分析，再来修正。如果改成 XXX_EXCEL 文件第 10 行名称为 XXX 的与第 28 行名称为 XXX 的身份证号码存在重复。这样就一目了然了，定位、修改错误也更方便。质检软件检查出的错误主要分为两大类，一类是矢量错误，例如界址点、界址线与地块边线不重合，或者地块存在重复节点等；另一种则是属性错误，例如编码错误、身份证号码重复、非空字段无数据等。

FME 有个特点是，按需定制，对于一些基本不会出现的错误，就没必要费时费力去做检查模板，只需要针对经常出现的问题，针对性的做出模板就好了。类似于点、线、面不套合的情况，只要前面使用 Snapper 转换器进行过处理，这种问题就不会出现，所以也没必要做这种检查。以下简单举例说明一下使用 FME 做检查的方法：

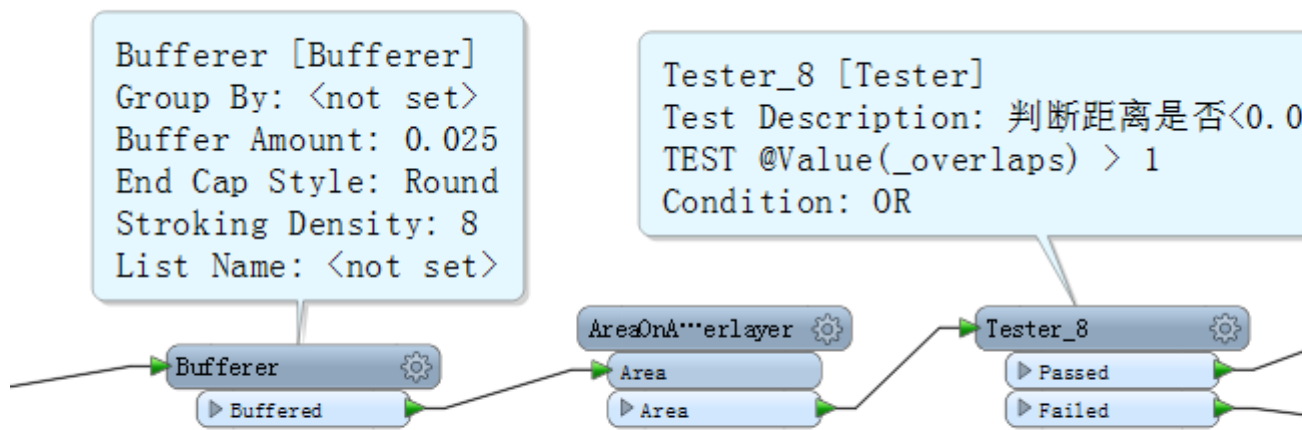
1) 点状要素重叠检查

关于点状要素之间距离不能超过 0.05 的描述，在质检规则里有几处，归根结底就是说，两点之间不论是否为同一地块，只要距离少于 0.05 就算错误，不论你这个点是界址点或者要素节点。由于 JZD、JZX 数据均由 DK 导出，三者是一致的，所以只需检查 JZD 一个表即可。

矢量 数据 检查	图 形 检查	点 状 要 素	点 状 要 素 重 叠 检查	要求同一图层中点状要素无相互重叠，即同一图层中点要素之间的实际距离小于0.05米时，应核实两个点状要素的真实性。	1、JZD 图层中的拓扑检查不符合质检要求的作为错误，其他点图层不符合质检要求的作为警告。。
		线 状 要 素	线 状 要 素 相 互 重 叠、 自 重 叠 检查	同一图层中线状要素无相互重叠、自重叠。	1、JZX 图层中的拓扑检查不符合质检要求的作为错误，其他线图层不符合质检要求的作为警告。
			线 状 要 素 自 相 交 检查	同一图层中线状要素无自相交。	1、JZX 图层中的拓扑检查不符合质检要求的作为错误，其他线图层不符合质检要求的作为警告。

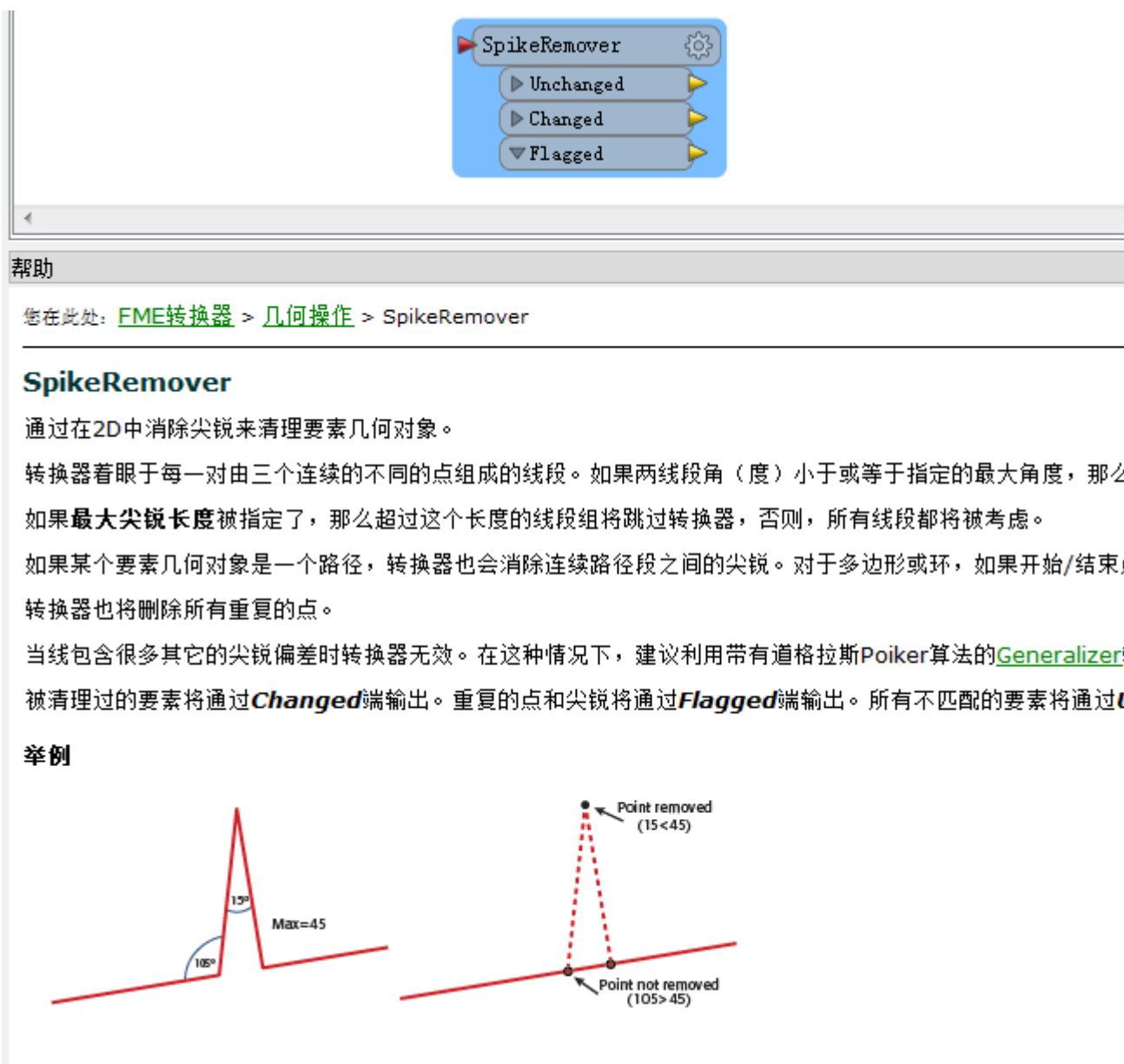
		线 状 要 素 碎 线 检查	承包地块界址线要素实地长度小于0.05米时，应核实界址线的真实性。	1、JZX 图层中的拓扑检查不符合质检要求的作为错误，其他线图层不符合质检要求的作为警告。
		面 状 要 素 节 点 重 复 检查	同一面状要素内节点间实地距离小于0.05米时，应核实节点的真实性。	1、DK 图层中的拓扑检查不符合质检要求的作为错误，其他面图层不符合质检要求的作为警告。
		面 状 要 素 相 邻 节 点 检查	同一面状图层内相邻要素的节点间实地距离小于0.05米时，应核实节点的真实性。	1、DK 图层中的拓扑检查不符合质检要求的作为错误，其他面图层不符合质检要求的作为警告。 2、检查同一面状图层中相邻要素的节点之间的实地距离，对于小于0.05米的，作为错误/警告。

检查方法是，将 JZD 缓冲一个 0.025 的圆，然后使用 AreaOnAreaOverlayer 做面叠加操作，最后判断 overlaps 的值，如果 >1 说明两个 0.025 的圆发生了相交，即点距离不到 0.05。



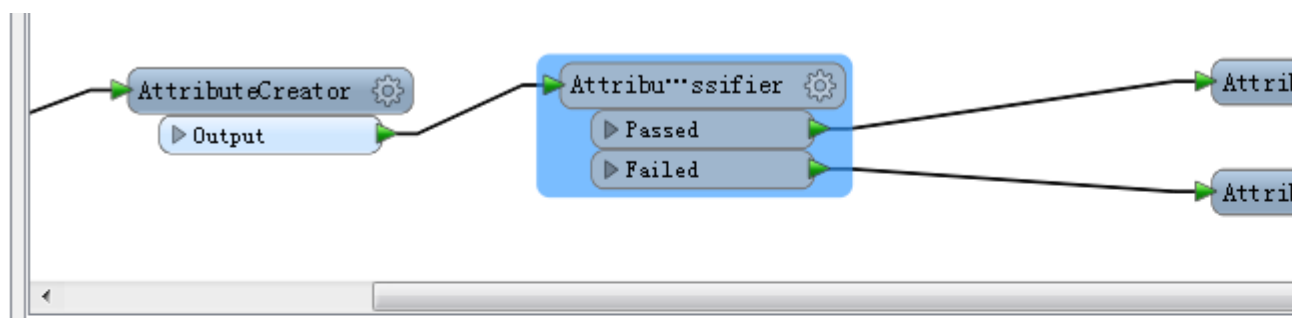
2) 地块狭长角检查

对于狭长角，建议要检查一下，我遇到很多这种情况，由于狭长角而导致的节点距离过近，而在图面上不细看，基本看不出，这种情况多出现在由 CAD 转成 SHP 格式的图件中。地块狭长角的检查也很简单，使用 SpikeRemover 转换器即可解决，该转换器说明如下：



3) 属性值是否符合要求的检查

数据库字段都有类型，是文本或是数字类型，如果不符，会导致该条记录入库不成功，所以检查属性值是否符合要求，也很必要，具体实现方法是使用 AttributeClassifier 转换器来判断属性值是否符合要求。该转换器说明如下：



帮助

您在此处: [FME转换器](#) > [字符串](#) > AttributeClassifier

AttributeClassifier

测试源属性的内容是否完全符合某个特定的字符类别，并输出相应的要素。

空字符串不能看成任意类别的一部分，所以总是会报错。

空的属性值仅与“Null”类型进行匹配，与其他类别不匹配。

输入端

▢ [Input](#)

输出端

▢ [Passed](#)

▢ [Failed](#)

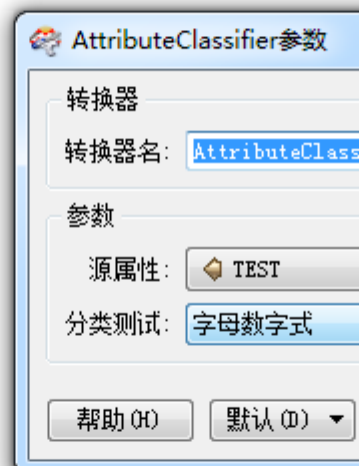
参数

▢ [源属性](#)

▢ [测试类别](#)

使用方法

该测试只有在指定具体值的情形下才能成功；任何非数字的值都不能成功。**NAN**是一个特殊的可以输出的值，如果您只想要找到值为非数字的属性，选择转换器**Double**，并使用转换器的 **Failed** 端口。



测试类别

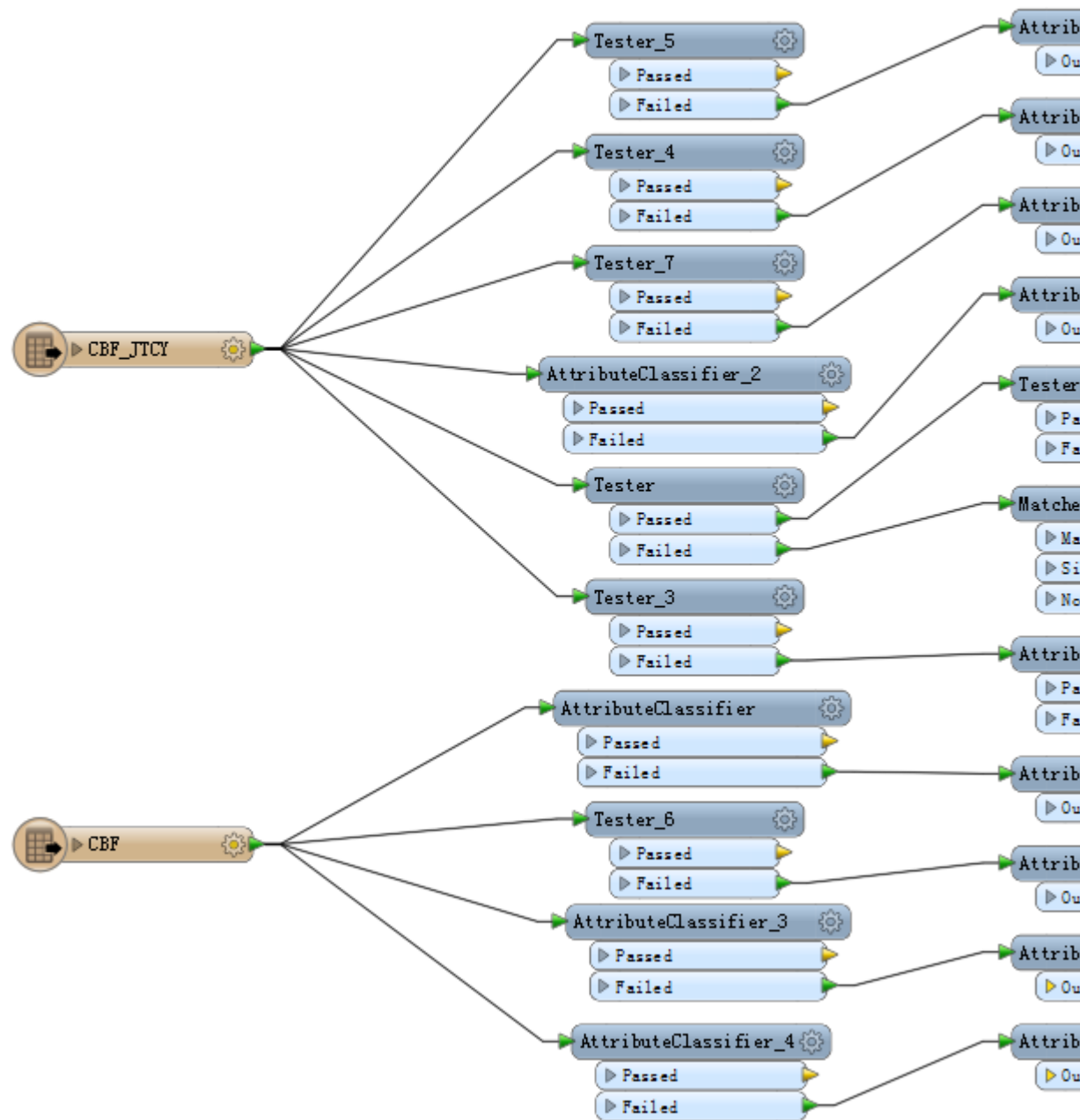
空字符串和错误的属性不能看成任意类别的一部分，所以总是会报错。

空的属性值仅与“Null”类型进行匹配，与其他类别不匹配。

下列字符类别可用：

字符类别	描述
字母数字	任意Unicode类型的字母或数字字符。
字母	任意Unicode类型的字母字符。
ASCII	任意一个值小于 \u0080（7位ASCII范围值）的字符。
布尔型	这个值可以是0, 1, false, true, no, 或yes。
控制	任意Unicode类型的控制字符。
日期型	任意格式为YYYYMMDD的日期。
数字型	任意Unicode类型的数字字符。注意，包括那些在0-9范围
双精度型	一个双精度的数值，包括：空格、符号标志、数字序列、小 任何值域都可以省略，小数点之前和之后的数字除外，另外
False	这个值可以是0, false,, 或 no。
字母	任意Unicode类型的打印字母，空格除外。
十六进制数字	任意一个十六进制数字([0-9A-Fa-f])。
整型	被定义为一个整型数字的组合，这个组合前面可以有符号，
小写字母	任意Unicode类型的小写字母字符。
NaN	一个浮点型的数字，它的值被指定为“非数字” 注意：该选项 正确使用的示例： 测试覆盖点到一个栅格高程模型。一些点 不正确的示例： 有一个属性代表了美国的邮政编码。您想要 AttributeClassifier来验证属性是否是整形。在其他情况下 也可以参考下面的使用方法。
空	属性值为空。
可打印的	任意Unicode类型的打印字符，包括空格。
标点符号	任意Unicode类型的控制字符。
空格	任意Unicode类型的空格字符。
True	这个值可以是1, true,, 或yes。
大写字母	Unicode类型字符集中的任意大写字母字符。
Wordchar	任意一个Unicode类型的单词字符。包括字母数字字符，Un

应该说，这个转换器测试条件算是比较全了，我们使用得较多的是判断是否为数字或者日期。判断字段是否为空，我一般使用Tester 转换器，使用 OR 把 Miss 及 Null 都包含在内。以下是我一个项目的查错模板，虽然看着检查项很少，实际上这几种错误是出现最多的，解决了这几种问题后，过质检软件没有大范围的错误了。



该模板使用了一个 Matcher 转换器，用来查找身份证号码重复，Matcher 既可以用来判断图形，也可以判断纯属性，通过该转换器后，Matched 端口出来的全是重复身份证号码，再排序一下，相同身份证号码的记录就在一起了。

4) 小结

每个人有每个人的作业习惯，每个单位也有每个单位的作业流程。同样，错误也有他的规律，我们通过错误查找规律来定制模板，针对性更强，实用性更好。有些错误理论上是存在的，可实际上一个项目可能永远也碰不到。这种情况我们就可以不考虑他，可以将精力留下来，做些更有意义的事情。使用 FME 做数据检查的目的，是要将一些对后期影响较大的瑕疵提前消除，例如数据有效性问题导致数据入库不成功等，如果后期改也不是很麻烦的小问题，那就交给质检软件去做吧。

七、 结束语

农经权做了两年多了，使用 FME 也做了不少关于农经权的模板，从调查阶段使用的属性挂接、四至求取、地块示意图生成、公示表格生成到现在数据并库、数据质检等等，两年下来，深有感触，FME 可以贯穿整个农经权作业流程毫无压力，而且人力投入没有其他软件那么大，模板调试起来也比较方便，学习难度对于那些编程软件应该说还是简单一些的，说到学习，真应该推荐一下 FME 中国技术交流群(群号 43814136 已满)，FME 中国技术交流群 2 (438653305)，在群里认识不少志同道合的人，也学到了很多有用的知识。非常感谢世纪安图提供的平台。

虽然写了这么多，连我自己都觉得用 FME 也没做啥高大上的东西出来，仔细想想，其实每个 FME 模板都很简单，例如全库 BSM 重编，比如点、线、面拓扑信息的建立，都只有几个转换器就完成了，批处理的部署，几行 BAT 命令就解决了。而这正是 FME 方便之处，把时间

留在好的思路中，而不是浪费在那些冗余的重复劳动中。以上是本人使用 FME 过程中的一些想法，如有不到之处，欢迎大家批评指正！