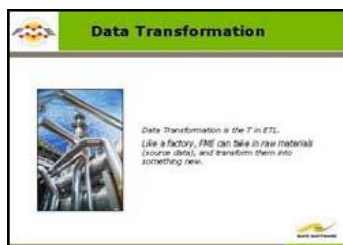


目录

数据转换.....	3
什么是数据转换?.....	3
数据转换类型.....	3
有关模式的概念.....	4
什么是模式?.....	4
FME 是怎样表示模式的?.....	4
在 FME Workbench 中查看模式.....	5
模式编辑.....	6
什么是模式编辑?.....	6
模式映射.....	7
什么是模式映射?.....	7
在 FME Workbench 中进行模式映射	8
FME Workbench 中的要素映射.....	8
FME Workbench 中的属性映射.....	8
内容转换.....	11
什么是要素?.....	11
“粗管”转换.....	11
几何特征转换.....	12
什么是几何特征转换?.....	12
属性转换.....	13
什么是属性转换?.....	13
使用函数进行转换.....	14
什么是函数?.....	14
函数的参数.....	15
函数颜色.....	16
函数端口.....	16
连续地使用函数.....	17
Feature Count Display – Serial Transformers	19

保存模式映射.....	20
Schema Mapping with Transformers	20
使用 FME Workbench 来检查数据.....	21
Connect Visualizer.....	21
放置一个 Visualizer	21
快速连接.....	23
什么是快速连接?.....	23
什么时候使用快速连接.....	23
平行地使用函数.....	24
Feature Count Display – 平行的函数	26
函数 Drag-and-Insert	27
插入多个端口的函数.....	28
对要素类别进行拖动和连接.....	28
基于组和基于要素的转换.....	29
基于要素的转换.....	29
基于组的转换.....	30
进行成组操作.....	31
坐标系转换.....	33
坐标系基础.....	33
坐标系设置.....	33
单元复习.....	35
从这单元中你应该学到什么?	35
回答问题.....	36

数据转换



数据转换指的是 ETL 中的 T（转换），依赖其重构数据的能力，这样就能输出所需要的数据，甚至一定程度上说，输出的数据要优于输入的数据。

什么是数据转换？

数据转换是 FME 重构数据的功能。这个功能的一个重要特征就是，用户不需要手动操作，就会自动地在读取（提取）和编写（加载）数据之间进行转换。

数据转换类型

数据转换又可以被具体分为两种：

结构转换

这类转换被称作“重组”可能更加合适，它是 FME 的一种功能，以近乎无穷的重组方式将源数据发送到目标数据，这个过程就包括，合并，划分，重构数据，以及自定义数据结构。

通过处理和操控数据集结构模式，就能够转换数据集结构。

内容转换

这类转换也被称作“重构”，它指的是改变一个数据集内容的功能；操控要素的几何特征或属性值就是关于如何使用 FME 来进行内容转换的最好例子。

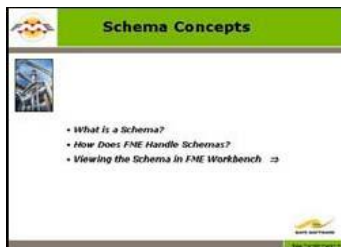
可以独立进行内容转换，也可以与模式转换一起进行。



Mr CAD 曾说过...

“和许多人一样，我购买 FME 仅仅是为了进行格式转换，但是数据转换工具给了我其它的选择！现在我使用 FME 来转换数据，即使有时并不需要对格式进行转换！”

有关模式的概念



转换一个数据集的结构就需要对模式有所了解，并且知道怎样使用 FME 来处理这些模式。

什么是模式？

模式指的是一个数据集的结构，或者更准确地说，它是数据集结构的正式定义。你可能比较熟悉术语 Data Model（数据模型），但是在 Safe Software 中，我们将它称为‘schema’。

每个数据集都有它自身独特的结构（模式），包括要素类别（层），几何特征，用户自定义属性，以及其它用来定义或限定内容的规则。你可以将它称之为“物理”模式，因为它表示的是数据的物理特性。

FME 是怎样表示模式的？

当你创建一个新的工作空间时，FME 就会扫描源数据集，从而就创建了一个源模式的工作空间定义，复制工作空间就能操作目标模式。

右图：在工作空间中，正如最开始显示的一样，源模式和目标模式，源数据在左边，目标数据在右边

每一项都是一个单独的要素类别，每个要素类别拥有一组属性

ADDRESSES [...]
ADDRESS
STREET_NAM
STREET_TYP
COUNTY
ZIPCODE
ZONING
INPUT_DATE
INPUT_BY
STAT
COMMENT_
PRIMARYINDEX

ADDR_TABLE ...
ADDRESS
STREET_NAM
STREET_TYP
COUNTY
ZIPCODE
ZONING
INPUT_DATE
INPUT_BY
STAT
COMMENT_
PRIMARYINDEX

这种情况下，源数据集模式表示的是“我们所拥有的”，而目标数据集模式通常表示的是副本。仅仅当要求对目标格式进行限时时，才会产生不同。

在这样的情况下，FME 就会尽力修复源模式和目标模式之间的差别，例如，当目标模式是 ESRI Shape 格式时，FME 会将所有属性名缩短为十个字符，这是因为 Shape 格式不支持大于十个字符的属性名。

为了更加专业，目标模式也被叫做“逻辑”模式，因为实际上来说它们并不是实际存在的。

在 FME Workbench 中查看模式

但是, 有时候我们并不能在工作空间中找到一个模式。模式的其它组成部分与整个数据集相关, 例如, 数据集所拥有的要素类别就被认为是模式的一部分, 并且在 Workbench 窗口中对它进行了描述。

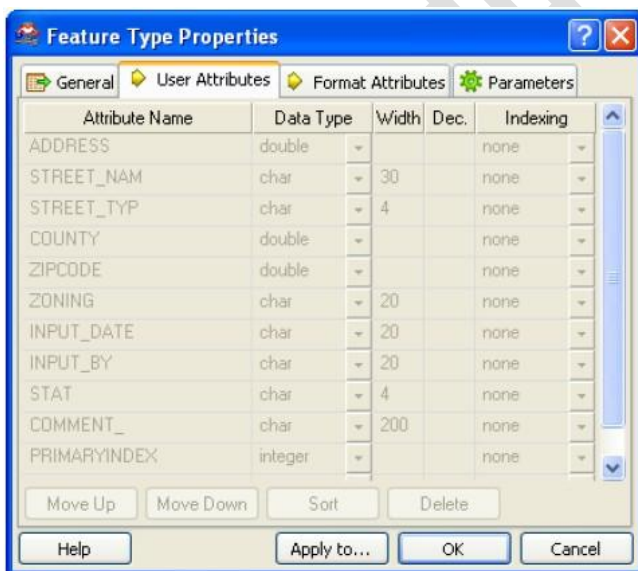
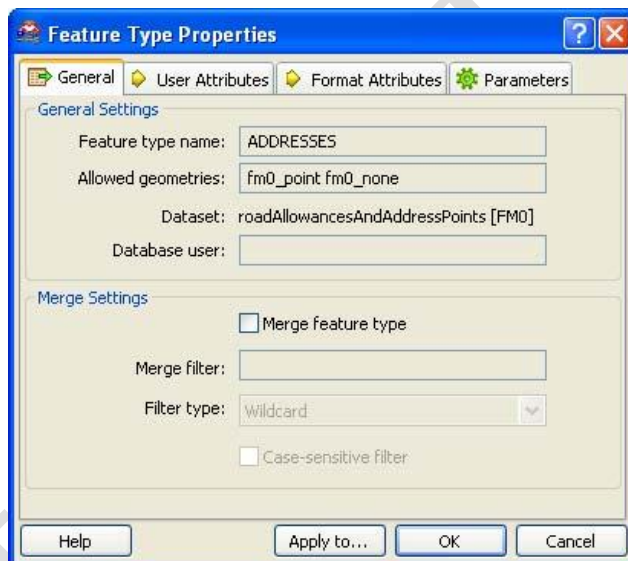
模式的一些部分是仅仅与一个单一的要素类相关的, 属性就是这样的部分。这些组成部分显示在 Feature Type Properties 的对话框中。

使用要素类别右边的[...]按钮就能够打开这个对话框。

右图: Feature Type Properties 对话框含有大量的制表符, 来显示信息。

在“General”这一栏显示的是要素类别名, 这时, 就要填 “ADDRESS”。

这里也显示了允许的几何特征类别, 它是要素类别的父数据集名。



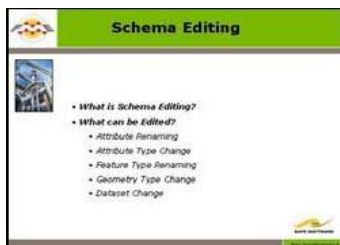
左图: ‘User Attributes’一栏显示的是属性类表, 根据属性名, 数据类别, 宽度, 以及小数位的个数, 这些因素来对每个属性进行定义。

这个例子中出现了一个源要素类别。默认地, 不能对源属性进行编辑, 这是因为源属性表示的是数据的物理性模式。如果改变了源属性, 模式就不能与源数据集相匹配了。



The ‘Data Type’ column for an attribute shows only values that match the permitted types for that data format. For example, an Oracle destination schema permits attribute types of varchar or clob. MapInfo does not support these data types so they will never appear in a MapInfo schema.

模式编辑



FME 创建的默认模式非常适合进行“快速转换”。当你需要自定义输出模式时，就可以使用 Workbench 对模式进行编辑。

什么是模式编辑？

最开始，我们将工作空间创建成目标模式，也就是源模式的副本。但是，很多情况下，用户会要求不同的输出数据模式。

模式编辑指的就是，改变目标模式，对输出的数据结构进行自定义。对输出数据的属性进行重新命名，就是一个很好的例子。

这种情况下，源模式仍然表示的是“我们所拥有的”，而目标模式则表示“我们所需要的”。

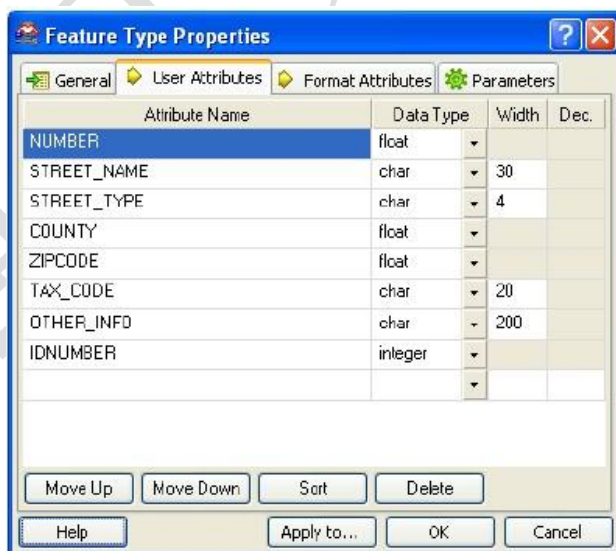
可以对那些内容进行编辑？

我们可以对很多内容进行编辑，包括如下：

属性重新命名

目标模式中的任何属性都可以被重命名，例如，重新命名 ROADS_ID to IDNUMBER。

打开 Feature Type Properties 对话框，点击‘User Attributes’（如右图），然后点击你要重新命名的属性，最后输入新的名称。



属性类型更改

目标模式中的任何属性都能改变原有的类型，例如，将 ID 的整型数改为小数类型

打开 Feature Type Properties 对话框，点击‘User Attributes’，使用 Data Type 字段就能改变属性类型了。

要素类别重命名

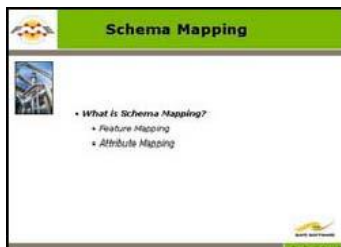
要对一个目标要素类别进行重新命名，例如，将 roads 重名为 updated_roads，就只需要打开 Feature Type Properties 对话框，点击‘General’，然后点击要编辑的要素类别名称字段，就可以编辑新的名称了。

几何特征类型更改

要对一个要素类别的几何特征进行更改，例如，将线类几何特征改为点类，只需要打开 Feature Type Properties 对话框，点击‘General’，从几何特征列表中选择你需要的类别。

备注：只有当格式要求一个几何特征类别时，这个字段才有效。

模式映射



模式映射就是将数据集的源模式与目标模式对应起来的一种方法，它是重建数据结构的基础

什么是模式映射？

在 FME Workbench 中，工作空间的一边显示的是源模式，另一边显示的是目标模式。模式映射就是用某种方法将源模式和目标模式连接起来，并且确保源要素与目标要素类别，源属性与目标属性是分别对应的。

要素映射

他指的是将源要素类别和目标要素类别连接起来。

属性映射

它指的是将源属性与目标属性连接起来。



Ms. Analyst 曾说过...

“你可以将模式编辑和映射理解为重构数据，这和整理衣橱有点像：扔掉你不需要的东西，为打算购买的东西保留空间，将已有的东西移到更加有用的位置。”

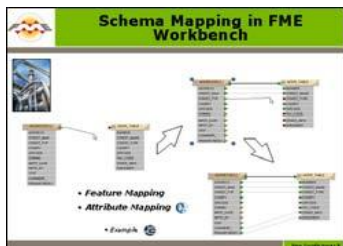
在 Workbench 自身的界面中，将模式的要素类别和属性进行连线，就能够将它们连接起来了。

域映射

一些格式支持域，也就是说，用一种方法来决定特殊属性需要的值。对于编码过的域来说（每个值都对应一个代码），有时候，要求对它进行映射，将数字型的值转换为描述性文字的值。

当需要改变一个域到另一个域时，也需要进行域映射。

在 FME Workbench 中进行模式映射

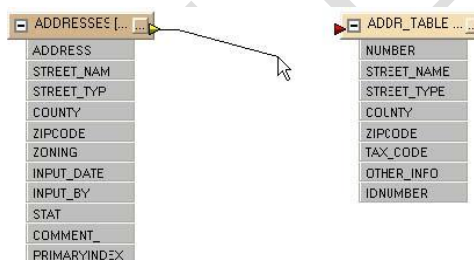


大多数情况下，在一个新的工作空间中，FME 会自动进行模式映射，但也可以按照要求编辑模式映射。

FME Workbench 中的要素映射

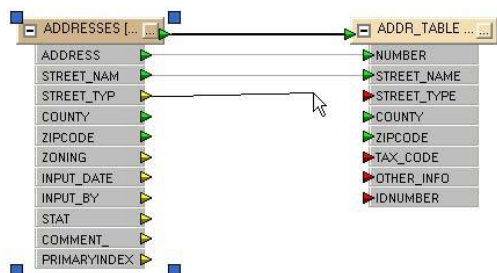
点击源要素类别的输出端口，按住鼠标将箭头拖到目标要素类别的输入端口，然后放开鼠标，就可以了。

右图：如右图所示，进行要素映射。



FME Workbench 中的属性映射

点击源属性的输出端口，按住鼠标将箭头拖到目标属性的输入端口，然后放开鼠标，就可以了。

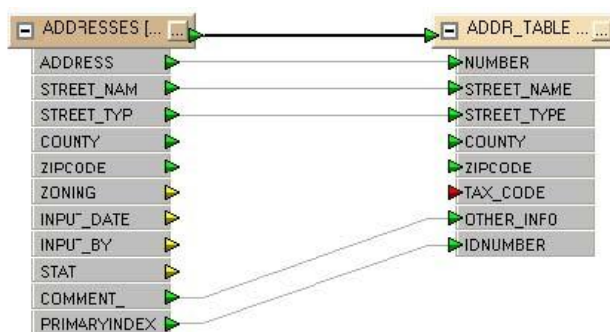


左图：已经进行了要素映射，并且已经建立了属性连接。

注意，绿色，黄色，红色的箭头分别表示哪些是已经连接了的属性。

右图：将源要素类别连接到了目标要素别。

黑粗的线表示要素映射，而细的灰色的线则表示属性映射。



具有相同名称的源属性和目标属性会自动连接，属性字段名是区分字母大小写的，所以‘ROADS’

与‘roads’ 或‘Roads’是不同的。有时候，不同名称的属性也会映射，但是却没有连接箭头显示，所以是看不到的。

绿色箭头表示已连接的属性，黄色表示还未与目标属性相连接的源属性，红色表示还没连接到源属性的目标属性。

例 1：模式映射

City_parks 是有关一个城市公园的一个 MapInfo TAB 的数据集，要对它进行模式映射

为了更准确地估算地面维护成本（割草，修树），你就需要全面处理数据，然后计算出每个公园的平均面积

首先要做的就是启动转换，并且确保模式映射能够运行

1) 启动 Workbench.

运用 New Workspace 对话框，使用以下信息创建一个工作空间：

Source Format	MapInfo TAB (MFAL)
Source Dataset	C:\FMEData\Data\Parks\city_parks.tab
Destination Format	MapInfo TAB (MFAL) (yes – here we write back to the same format!)

2) 调整目标模式

当目标映射和源映射相符合时，FME 会自动创建一个工作空间，但是，在这个操作中，我们想赋予目标属性更多有意义的名字，删除一些不需要的属性，添加需要的新的属性

点击 properties 按键，或者右击要素类别，选择“Properties...”，就能够打开目标要素类别的 Feature Type Properties 对话框

点击 User Attributes 标签页，就会出现目标属性的列表

- 重新命名字段 name 为 ParkName
- 重新命名字段 name_alt 为 ParkNameAlt
- 添加一个新字段 ParkArea [make sure it is attribute type of FLOAT]
- 添加一个新字段 AverageParkArea [make sure it is attribute type of FLOAT]

点击 OK，同意这些更改

3) 映射目标模式

现在目标属性名已经改变了，FME 就会终止之前的所有模式映射，这时你的工作空间如右图所示。



重新映射公园名和可替换的名字到对应，相等的目标数据模式中，这时你的工作空间如右图所示。



注意，现在已经没有任何属性要映射到 **ParkArea** 或 **AverageParkArea** 中了，因为它们对应的源数据值并不存在，但是，你仍然可以运行工作空间，查看输出状态。

不要忘记保存这个工作空间，因为之后你可能会用到它。



Miss Vector 曾说过...

“同学们！现在又要进行一个小测试了，来检查你们学习的成果。问一下其他的成员，然后相互回答这些问题。”

“ETL” 中的 “T” 代表的是：

- 1) Translation
- 2) Transfiguration
- 3) Transmission
- 4) Transformation

一个模式的源组成部分和目标组成部分分别代表的是：

- 1) What we have/What we want
- 2) What we did/What we should have done
- 3) What we're adding/What we're removing
- 4) Where my data was/Where it is now

将源模式连接到目标模式，我们将这个过程叫做：

- 1) Schema Mapping
- 2) Schema Connecting
- 3) Schema Joins
- 4) Schema Concatenating

下面的那一种箭头颜色表示 FME 模式中没有进行连接？

- 1) Red
- 2) Green
- 3) Blue
- 4) Yellow

内容转换



内容转换指的是，转换一个数据的几何特征或属性。

什么是要素？

FME 中的要素在转换过程中是一个个体，例如，GIS 或制图要素，除了一组相关的属性外，还包括几何特征。FME 能够对属性或几何特征进行重构



FME 中的要素是数据的最基本的（亦即最小的）单位

FME 中的要素表达法是灵活的，通用的，也就是说，对这些要素有一个基本的 FME 定义，与要素所在格式无关



几何要素
(Geometry)



数据库记录
(Database Record)

“粗管”转换

“粗管”转换描述了 FME 的内容转换功能，也就是说，你可以在转换过程中处理数据，然后产生一个目标数据集，并且这个数据集的数量会多于源数据集的



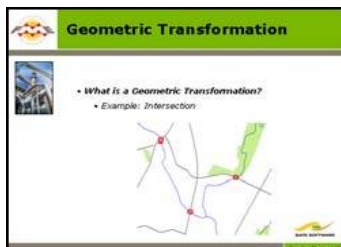
上图：注意观察属性 A,B,C 是怎样转换成一组新属性—W,X,Y 和 Z 的。箭头表示转换过程是双向的。



Mr Flibble 曾说过...

“现在是挑战你的时候了！根据用户反馈意见，最常见的数据转换是将 ESRI Shape 转换为那种格式的数据？如果你听了课的话，就请告诉我拟的答案。”

几何特征转换



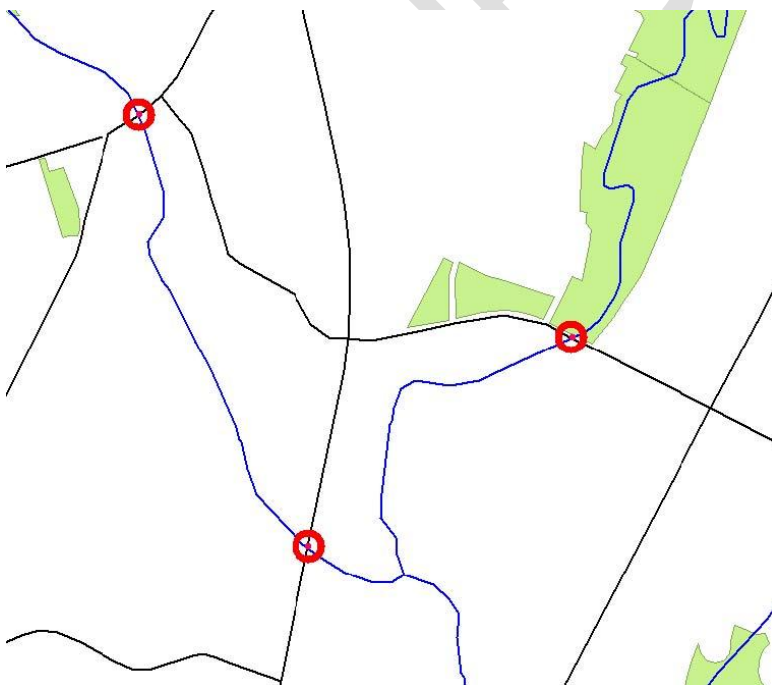
几何特征转换是一个属于，指的是在 FME 中对于一个要素的空间（或几何）部分进行结构重构

什么是几何特征转换？

它指的是对 FME 要素的空间组成部分进行结构重构，换句话说，就是改变要素的几何特征，然后输出不同类型的要素。

下面是一些有关几何特征转换的例子...

- 缩编——一种制图过程，通过重构数据结构，使其在特定的比例尺的地图中都能轻易地查看到数据。
- 仿射——调整要素的大小和形状，使其更加匹配参考数据
- 拓扑计算——将一组线性要素转换为点/线结构



左图：一个有关线—线交汇的例子。

在公路和河流交汇的地方就会产生一个点，表示桥的位置。

属性转换



属性转换是一个术语，指的是在 FME 内，对一个要素的非空间（或属性）部分进行结构重构

什么是属性转换？

它指的是，对一个 FME 要素的非空间部分进行结构重构，换句话说，就是改变属性的物理几何特征，然后输出不同类型的要素。

下面就是属性转换的一些例子：

- 关联-将两个或多个要素结合起来
- 分拆- 缩编的相反操作，将一个属性分拆为多个
- 测量 - 测量一个要素的长度或范围，创建一个新的属性
- 创建 ID - 为特殊的要素创建一个特定的 ID

	Address1	Suite 2017,
+	Address2	7445-132nd Street,
+	City	Surrey,
+	Province	British Columbia,
+	PostCode	V3W 1J8
=	Address	Suite 2017, 7445-132nd Street, Surrey, British Columbia, V3W 1J8

上面：属性关联是有关属性转换的一个例子。将地址的每一行进行并别，然后发送回一个只有一行的地址。

使用函数进行转换



你已经知道了，在工作流中，我们用对象来表示 FME Workbench 的源数据和目标数据，同样地，我们用对象来描述几何和属性重构过程—函数

什么是函数？

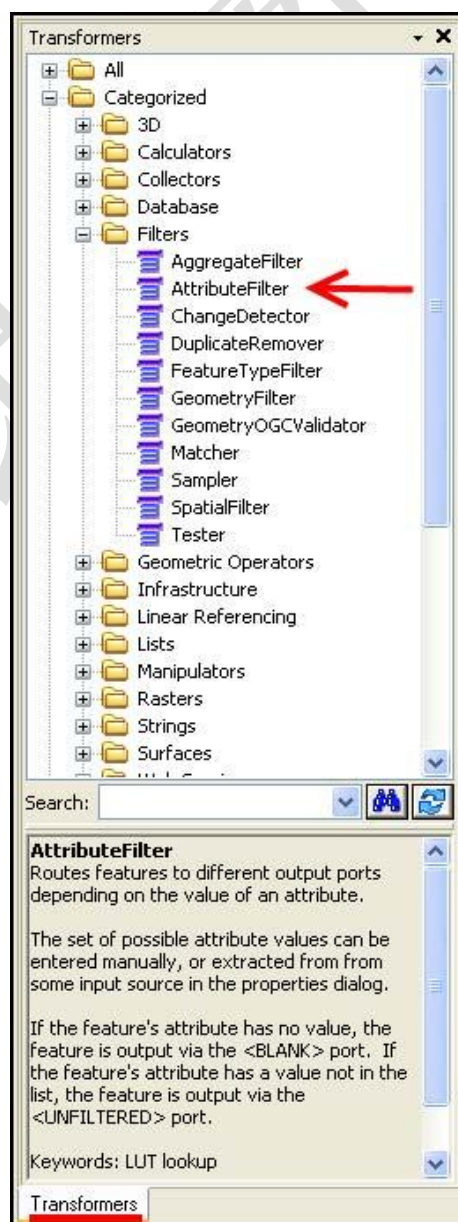
函数就是 FME Workbench 的一个对象，它执行要素重构的任务。FME 存在大量的函数，用来进行不同类型的结构重构。

Workbench 的显示窗口可能不同，但是一般它的导航窗口都会有一个标签—‘Transformers’。点击它，查看 FME 的函数列表。

为了方便用户查看函数，我们将 Transformer Gallery 分为不同的单元和类型。

在它的下方也有一个选项，它提供了已选函数的基本信息。

右图：已经打开了一个特定的函数（Filter），并且高亮显示了 AttributeFilter 函数。

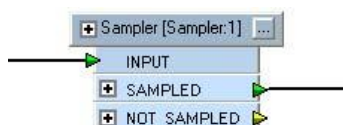


放置一个函数

一旦你选中了一个函数函数列表中，双击它，就可以将这个函数放置到工作空间中了。

或者，你可以将函数拖到工作空间中。许多用户选择这种方式，这是因为通过拖动函数，能让你更好地控制函数的初始放置位置

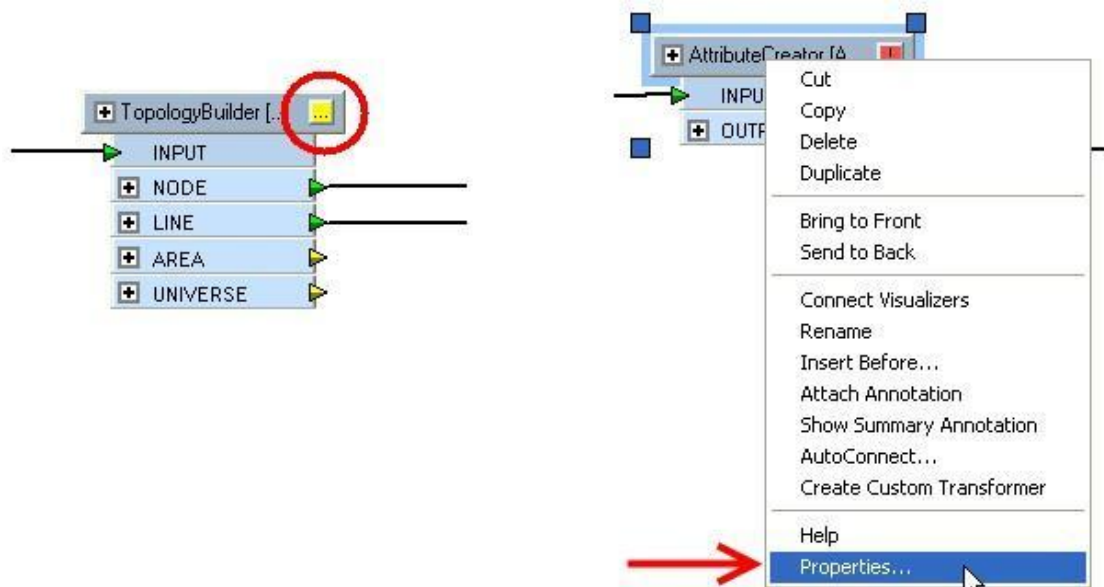
在放置了函数后，将输入端口连接到输出端口，就可以添加函数了。



上图：放置和连接函数的例子

函数的参数

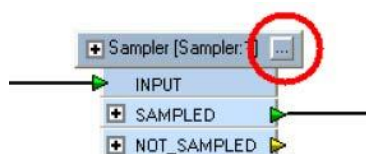
每个函数可能有许多参数—也叫做设置。通过点击函数右上方的 properties 按键，或者右击一个函数，然后选择‘Properties’选项，就可以访问参数了。



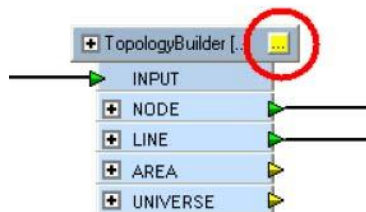
颜色含义

Properties 按键的颜色，如下图

一个函数的 properties 按键被标记成不同的颜色，表示设置目前的状态



蓝色：表示已经检查和更改了默认函数设置，并且即将使用这个函数。

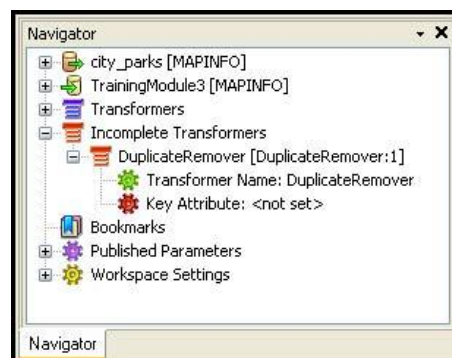


黄色：表示还没有检查默认设置，可以使用这个函数，但是结果可能是错误的。



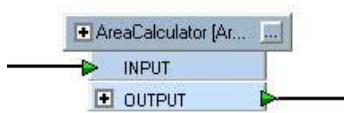
红色：表示至少有一个默认值 FME 无法提供，在使用函数之前，一定要由用户提供一个设置值。

右图：不完整的函数（带有红色标记），并且也会在导航方框中显示。



函数颜色

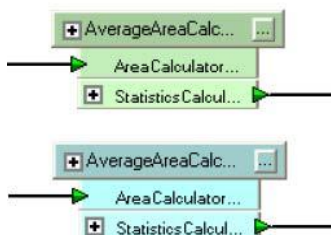
函数显示成不同的颜色



一般，所有常用函数显示为蓝色，比如，AreaCalculator 函数（左图）

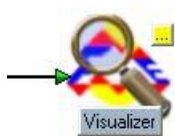
自定义的函数有两种形式：嵌入或

右图：嵌入的自定义函数显示为绿色
备注：如果你不知道自定义函数是我们会详细讲解。



连接。

色，而连接的函数显示为青色。哪一种类型，不要着急！下面

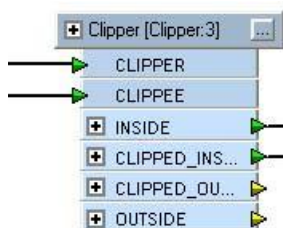
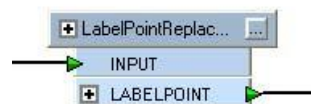


一些函数，例如，Visualizer（如左图），拥有它自己不同的图标。

函数端口

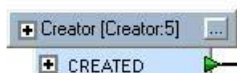
一个函数不仅可以拥有单一的输入和输出端口，也可以拥有多个输入，输出端口。

右图：这个 LabelPointReplacer 仅仅有一个单一的输入，输出端口。



左图：Clipper 拥有多个输入和输出端口，注意，这些端口都需要进行那个连接

右图：Logger 函数是特殊的，因为它只有一个输入端口。



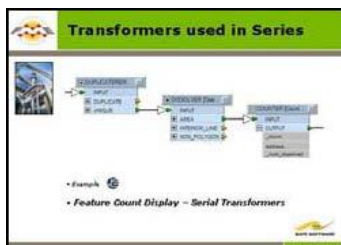
左图：Creator 只有一个单一的输出端口。



First-Officer Transformer 曾说过...

“如果你有一个红色标记的函数，导致整个转换过程无法完整运转。在进行转换之前，你需要修正这些错误。”

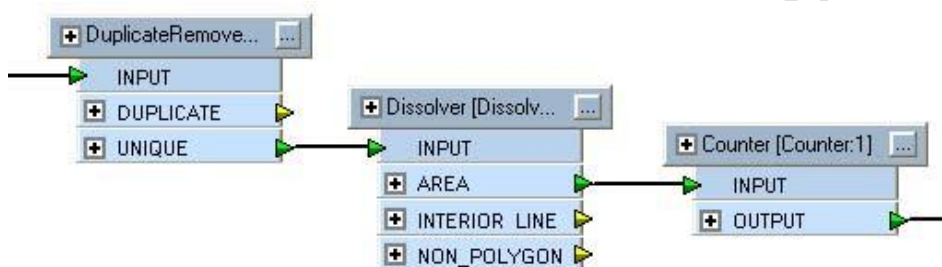
连续地使用函数



和电路中的组件非常相似，我们可以将连续的函数结合起来使用，这样一系列函数就可以对数据产生。

虽然可以在 FME 中获取大量的函数，但是通常用户会发现，单一的函数并不能满足要求。这种情况下，你就需要将多个函数结合起来使用。

FME 的一个重要概念就是，将一连串函数图形化地显示整个工作流



上图：这个例子中，DuplicateRemover 函数用来删除多余的多边形要素。在一个共有边界内，Dissolver 函数将每个唯一的多边形与它的相邻多边形融合起来。最后，每一个整合的区域就会从 Counter 函数中获取一个 ID 名。



例 2：基本的函数使用

在例 1 中已经使用了一些基本函数，来为公园测量创建转换工作空间，现在，我们会使用 FME 函数来测量那个公园的面积，并且计算平均值。

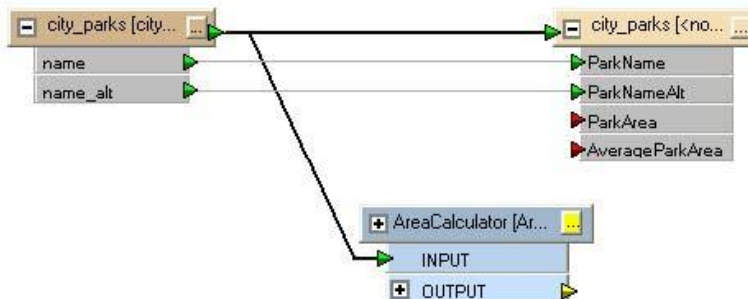
1) 启动 Workbench.

打开例 1 中的工作空间，或则会重新创建一个工作看空间。

2) 添加 AreaCalculator 函数

在函数一栏，放置函数 AreaCalculator，双击它，将它放入到工作空间。

点击源要素类别的输出箭头，然后拖动鼠标，将它连接到 AreaCalculator 函数的 INPUT 端口。



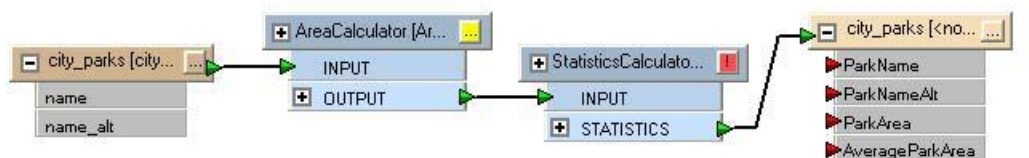
右图：这时候的工作空间如右图所示：

3) 添加 StatisticsCalculator 函数

与放置 AreaCalculator 函数的方法相同，找到并且添加一个 StatisticsCalculator 函数。将 AreaCalculator 的 OUTPUT 端口连接到 StatisticsCalculator 的 INPUT 端口。

将 StatisticsCalculator 的输出端口连接到目标 GML 要素类别。因为不再需要之前的连接，所有你可以点击连线，同时按键盘上的 DELETE 键，这样就可以删除它了。

这时你的工作空间如下。注意，之前的属性映射已经丢失了，但是不要担心，下面我们会进行详细讲解。



4) 检查函数设置

打开 AreaCalculator 函数的参数对话框，默认设置会将计算值放如到一个叫做_area.的属性中，但是，目标模式显示的是，要创建一个叫做 ParkArea 的属性，所以需要改变这个设置，然后创建正确的属性，亦即将_area 更改为 parkarea。

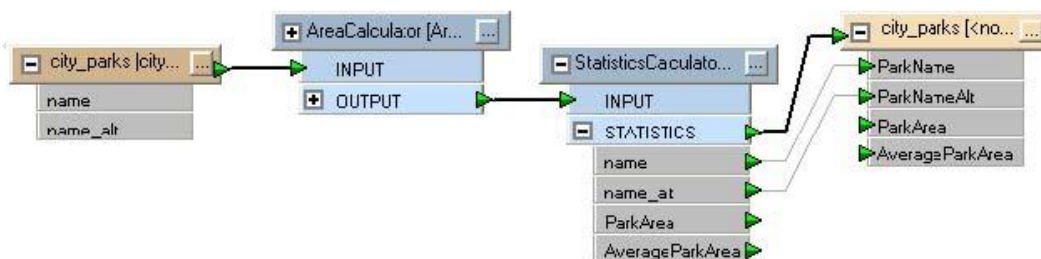
打开 StatisticsCalculator 函数的对话框，对包含面积的属性进行分析，也就是已经重新命名的 ParkArea 属性，一定要将“Pass Through Features?”设置成 YES。

我们需要检查哪些默认设置是用来计算公园的平均面积的。这个检查有必要吗？按照要求更改设置，使它符合目标模式。最保险的操作就是，不要对不需要的 StatisticsCalculator 属性进行任何设置。

5) 重新映射模式

为了完成设置，我们需要恢复最开始的属性模式映射。注意，我们在函数设置中使用的是正确的属性名，所以只需要映射 ParkName 和 ParkNameAlt；而 ParkArea 和 AverageParkArea 应该已经建立了连接。

按照要求，增加 StatisticsCalculator 的属性列表，进行属性连接。



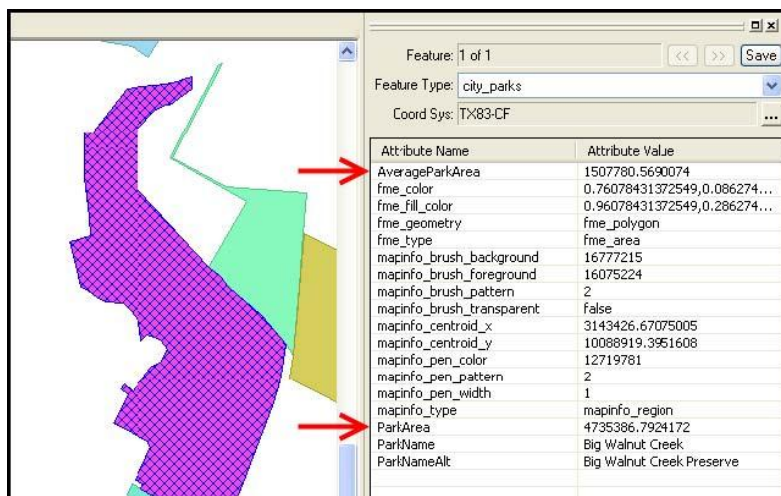
这时你的工作空间经过简单清理后，应如有右图所示。

不要忘记了保存工作空间，因为之后会用到！

6) 开始进行转换
按照提示，你会被要求输入
输出数据集。

检查转换结果。查询一个要素就会在属性列表中找到公园的平均面积。

右图：Universal Viewer 中的
输出结果。



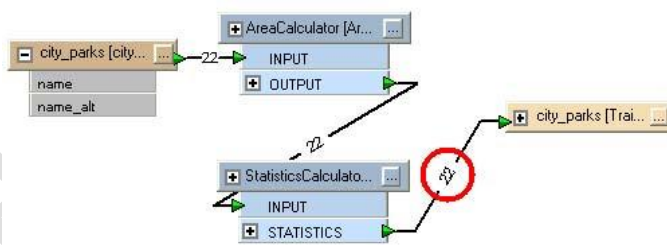
Attribute Name	Attribute Value
AverageParkArea	1507780.5690074
fme_color	0.76078431372549,0.086274...
fme_fill_color	0.96078431372549,0.286274...
fme_geometry	fme_polygon
fme_type	fme_area
mapinfo_brush_background	16777215
mapinfo_brush_foreground	16075224
mapinfo_brush_pattern	2
mapinfo_brush_transparent	false
mapinfo_centroid_x	3143426.67075005
mapinfo_centroid_y	10088919.3951608
mapinfo_pen_color	12719781
mapinfo_pen_pattern	2
mapinfo_pen_width	1
mapinfo_type	mapinfo_region
ParkArea	4735386.7924172
ParkName	Big Walnut Creek
ParkNameAlt	Big Walnut Creek Preserve

Feature Count Display – Serial Transformers

现在，我们来看之前数据转换的例子，你注意到了吗？在完成转换后，每个连接都标记了转换数据的数目。

右图：之前的例子

Feature Count 显示 22 个要素通过了
这两个函数，然后又被输出。



日志窗口会核查编写的要素数目。

Feature Count 显示有多少个要素通
过了每个函数。

如果目标数据与要求的数据不一样，这个功能就能帮助分析工作空间，并且提供调试意见参考。

之前的例子显示所有连接上的要素数目相同，但是，但是当你使用一些函数，就可能会影响到要素的数目，例如，DuplicateRemover 或 2DPointReplacer 函数。下面的例子会详细说明这个问题。

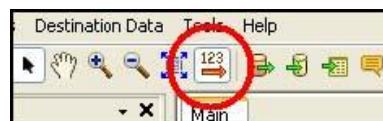


左图：Feature Count 显示有 22 个要素进入到了函数 DuplicateRemover，但是最终只有 18 个有唯一的属性值的要素被输出。

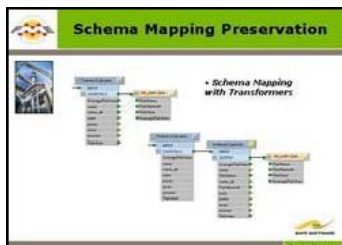


使用工具条中的打开，关闭按钮就能够执行 Feature Count Display 功能。

右图：FME 工具条中的‘Display Feature Counts’按钮



保存模式映射



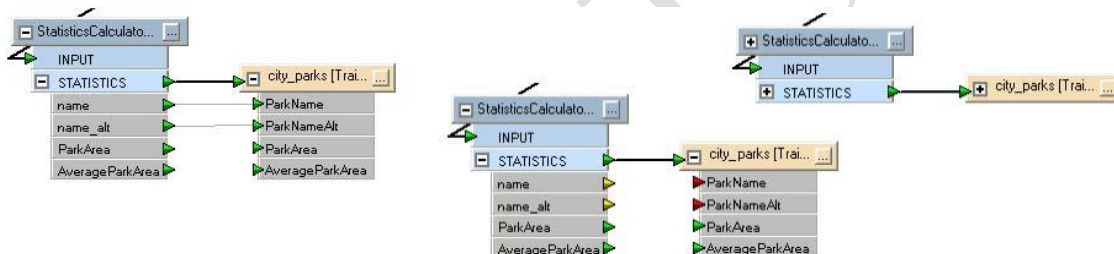
一些函数并不是用来进行数据转换，但是却可以帮助在工作空间中，通过模式映射来设置输出数据。

Schema Mapping with Transformers

虽然这个话题看起来有点偏题，但是它能够解决前面例子中出现的问题，所以值得关注，但是并不推荐使用手动属性映射。

在例 2 中，出现了一个问题，我们必须尽力解决：当你添加一个新的函数到发送的数据流中时，就会丢失掉最开始的属性映射。

当属性列表最小化时，手动属性映射就会成为出现弊端，因为你不能看到丢失掉的连接。

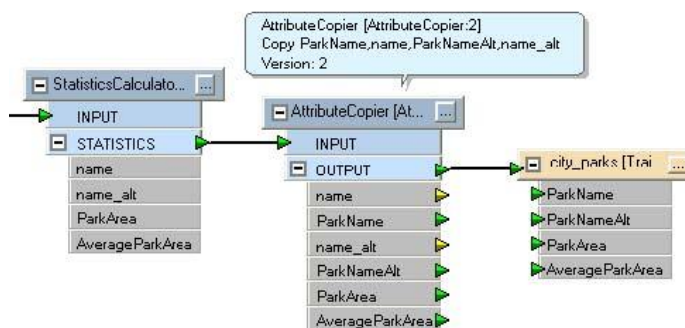


上图：当你扩展属性列表时，就能很轻易地查看是否属性映射进行了连接，而一个完整的列表是不能告诉你这些的。

进行属性映射的方法，并不是手动进行连接，而是对属性进行命名或重命名，使之能符合目标模式。

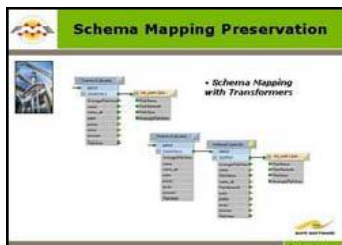
这项功能能够保证，属性能够自动地发送到目标模式中，例如，你会注意到，**ParkArea** 和 **AverageParkArea** 并不需要进行模式映射，这是因为在函数设置中已经自动地设置了与之符合的属性。

当你需要对源属性进行配对时，使用 **AttributeRenamer**（或 **AttributeCopier**）函数来进行转换，这样的话，添加其它函数就不会影响到对属性的映射了。



右图：用 **AttributeCopier** 来取代手动属性映射。

使用 FME Workbench 来检查数据

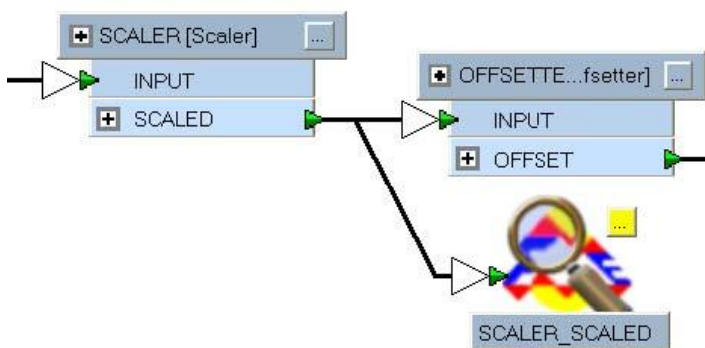


除此“Route to Visualizer”之外，FME workbench 能够从单个的函数发送数据到 FME Viewer。

Connect Visualizer

Visualizer 是一个 Workbench 函数，拥有它自己独特的形状和格式，输入其中的数据能够直接发送到 FME Universal Viewer 中。

Visualizer 不同于 Redirect to Visualizer 设置，它适用于转换过程的任何一个阶段，并且不会阻止输出数据到写入模块。而且，你促使用户更加细心地选择要查看哪些要素。



左图：这里，在选择数据后，就直接被发送到 Visualizer 中。

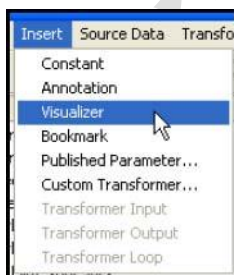
备注：甚至，你可以将 Visualizer 连接到源要素类别，在用 FME 读取数据后，就可以直接查看它们了。

放置一个 Visualizer

我们有许多方法来放置一个 Visualizer 函数。

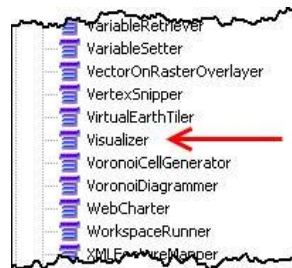
From the Transformer Gallery

Visualizer – 和其它函数一样，会出现在函数列表中（如右图），用相同的方法就能将它放置，连接到工作空间。



From the Menubar

使用 Workbench 菜单条（如左图）的 Insert > Visualizer，也能够放置 Visualizer。



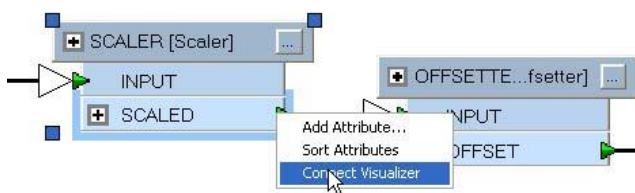
From the Toolbar

因为这个函数要经常使用到，所以在 **Workbench** 菜单条中可以找到一个特殊的图标（如右图）。



In the Canvas

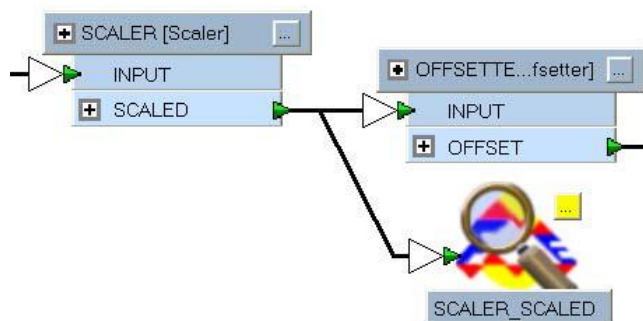
可能使用 **Visualizer** 的最简单方法就是，右击输出端口，然后选择‘Connect Visualizer’设置



左图：和之前的例子相同，用户需要将 **Scaler** 中的数据输出到 **FME Universal Viewer**。

右击“SCALED”，然后选择‘Connect Visualizer’。

右图：FME 添加 **Visualizer**，来连接 **Scaler** 函数。现在，被剪辑的数据就能直接发送到 **FME Universal Viewer** 中来进行检查。



注意是 **FME** 是怎样使用函数和输出端口名来自动命名 **Visualizer** 的



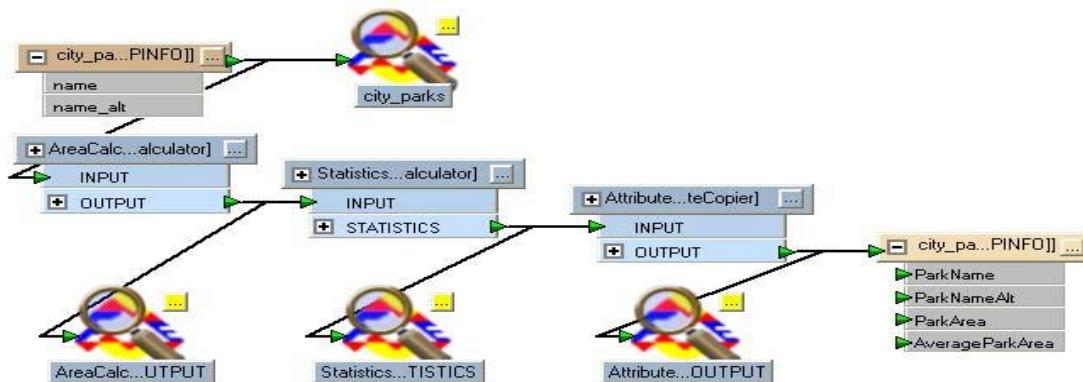
左图：查看器中的 **Display Control Window**。

观察，要素类别是怎样命名的，这与工作空间中的 **Visualizer** 方式一样。当使用多个 **Visualizers** 时，这个方式能帮助用户区分数据。

Mr E. Dict, (Attorney of FME Law) 曾说过...

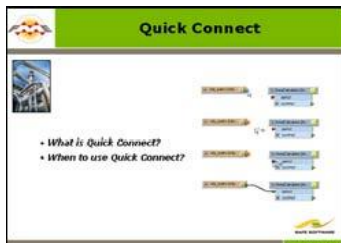


“按照培训课程的 4a-(27)安排，你需要重新打开例 2 中的工作空间，然后添加 **Visualizer** 函数到每一个函数对象上，来练习使用这项功能。”





快速连接

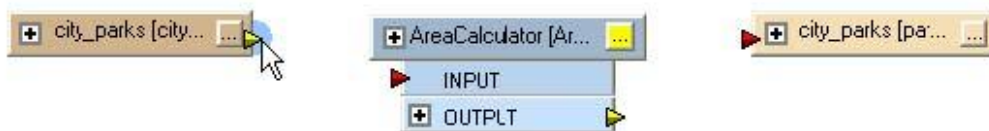


它是将不同的工作空间中的对象连接在一起的新方法。

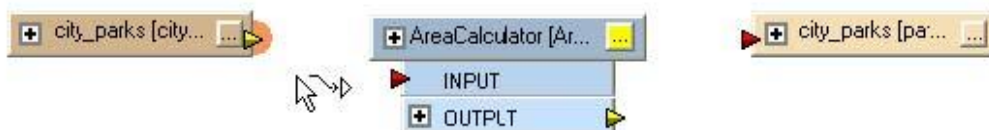
什么是快速连接?

它指的用一种方法，将函数和要素类别连接起来，但是不需要在它们之间划线，你要做的就是点击正确的输入和输出端口。

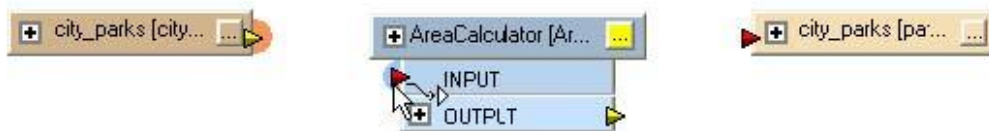
下图：点击输出端口，将它变为蓝色：



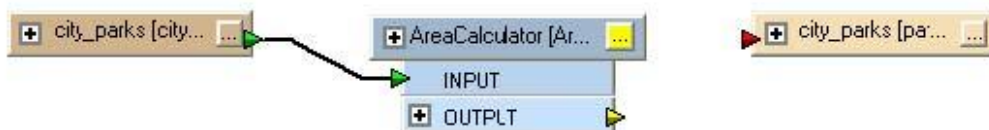
下图：点击这个端口，变成粉色，光标也会改变形状：



下图：点击一个输入端口，这个端口就会变成蓝色：



下图：点击端口，连接就完成了：

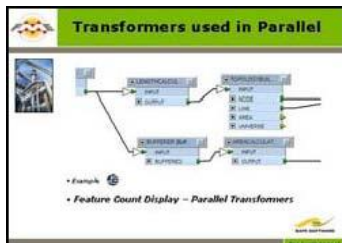


什么时候使用快速连接

当你需要对一个很大的工作空间中相隔很远的两个实例进行连接时，快速连接就是最好的选择。如果不使用快速连接，你就必须将这两个实例放大到能看到的尺寸，这样的话，进行连接就会比较困难。

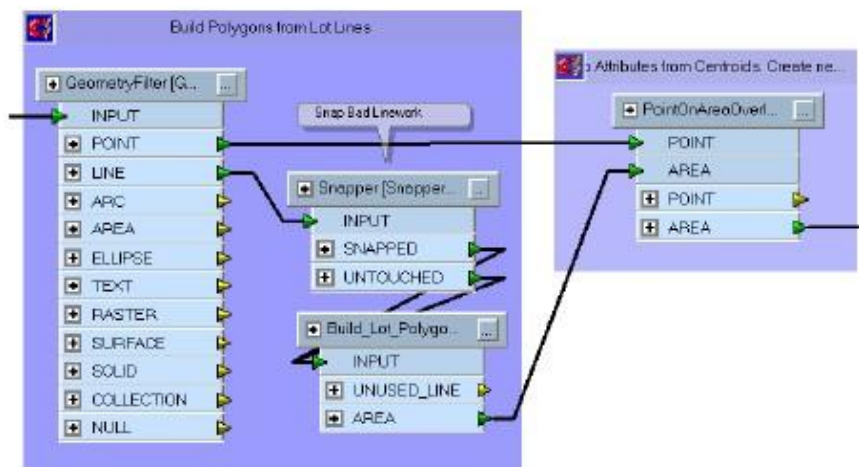
而使用快速连接，这两个实例就不需要同时显示在屏幕中，你可以点击第一个端口，然后使用窗口打包工具移到下一个窗口。

平行地使用函数



FME Workbench 允许多重数据流，而每个数据流都会各自流经自己的函数。

FME 的另一个重要功能就是，在工作流过程中将数据分拆为单独的数据流，或者将不同的数据流整合为一个单一的数据流。



左图：这个例子中，使用 GeometryFilter 函数将数据分拆为两种类别的数据流——点类和线类。

这样，就能够分别地对它们进行处理。

然后，再将这两种数据流合并为一个。

一定要记住，用这种方法分拆数据流会产生两份数据，其中一个副本会经过每个数据流。

例 3：并行使用基本函数



Example 2 使用 AreaCalculator 函数来测量公园面积，现在，按照要求我们要添加公园名称，对面积要素进行标记。使用 LabelPointReplacer 函数，FME 就能进行这项操作。

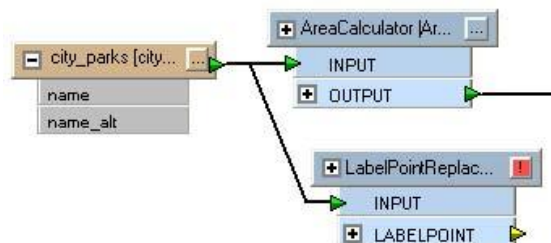
1) 启动 Workbench.

打开例 2 中的工作空间，或者再重新创建一个。

2) 放置一个 LabelPointReplacer 函数

放置一个 LabelPointReplacer 函数。

使用快速连接，将源要素类别和 LabelPointReplacer INPUT 端口进行连接（如右图）



3) 创建新的目标要素类别

右击目标要素类别，选择“Duplicate”，就能够在输出数据集中创建一个新的要素类别（层）
使用快速连接，将 LabelPointReplacer 连接到这个新的要素类别，你可以放心，不需要对这个新的连接进行任何属性映射。

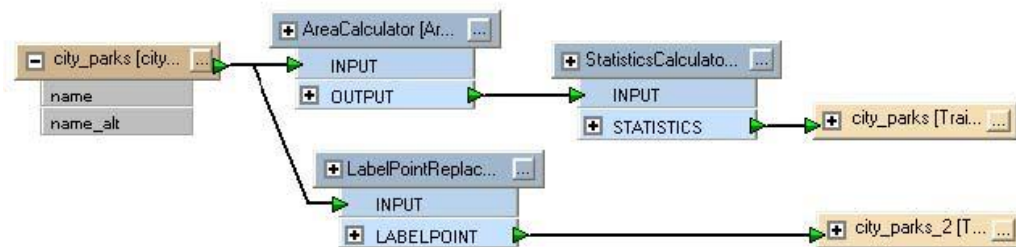
4) 检查函数设置

打开 LabelPointReplacer 函数的设置对话框

选择“name”属性为 Label 属性名，将 Label Height 设置为 500（尺）

Label Height 既可以是一个属性，也可以是一个常用值—其实可以输入字符/数据作为常用值，如果你不明白这点的话，就可能对 Label Height 感到很困惑。

工作空间如下图所示。



同样的，不要忘了保存工作空间，因为之后要用到它。

5) 开始转换

检查输出，如右图所示。

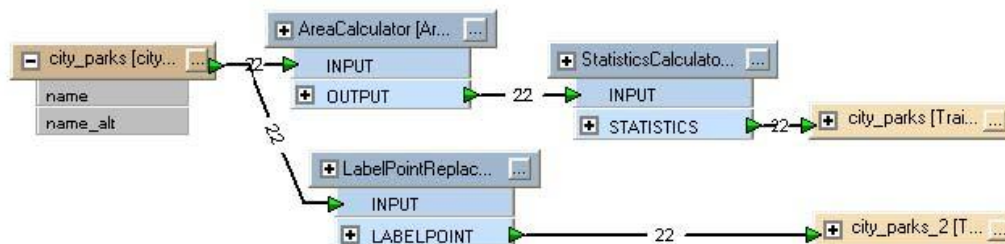
总结

在例 3 中，你从同一个输出端口（或源数据集）中，将数据分拆为两个数据流，一组完整的要素就会流经这两个数据流，也就是说，你同时处理了源数据和它的副本。



Feature Count Display – 平行的函数

前面例子中，在处理过程的最后，你是否注意到了 feature counts 函数？记住，Feature Count Display 显示的是流经了多少个要素

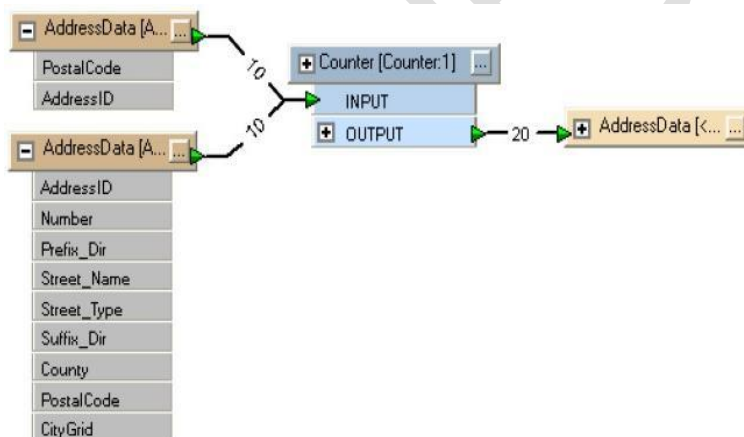


Above: 前面例子中，Feature Count 显示的是，当我们将数据分拆为两个单独的数据流时，就会额外产生一个数据副本，也就是说，22 格要素流经了数据流，而不是 11 个要素。虽然前面的例子展示了，用户怎样自己分拆数据，但是也可以使用一些函数来分拆数据，并且，并非所有函数都会产生一个要素副本。



左图：22 个要素流入 DuplicateRemover，22 个要素流出这个函数，但是，输出要素被分为两流，一个是 4 格要素，一个是 18 个要素。

Feature Count 显示了在工作空间中分拆要素会产生结果，同样的，它也可以显示合并要素流所产生的结果。

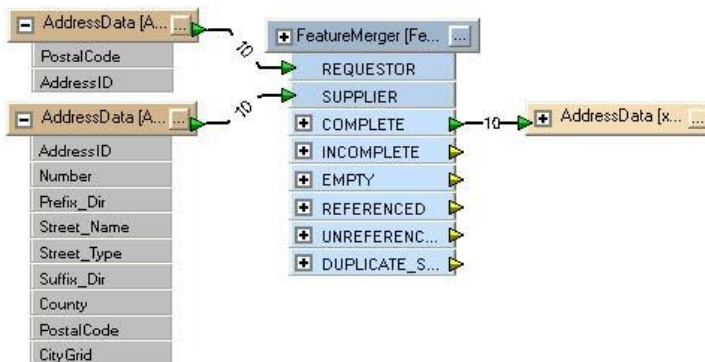


左图：尝试将一组空间要素和相关的数据库记录结合起来。

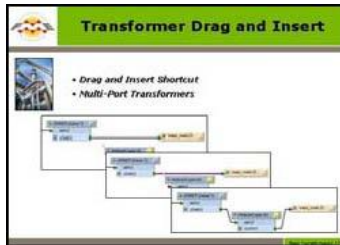
这个例子中，feature counts 表示，两个数据流的合并并不会产生合并。1 同一个源数据流中的 10 个要素只是简单地合并为 10 个要素流，也就是说，产生 20 个输出要素，10 个空间和 10 个非空间要素。

右图：使用 FeatureMerger 函数能更好地将空间和非空间数据结合起来。

在这个例子中，将每个数据流中的 10 个要素融合，然后输出 10 个要素。



函数 Drag-and-Insert



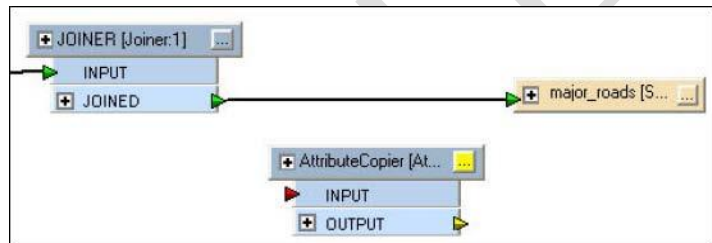
用户有必要了解怎样手工地连接不同的函数，同样地，也需要了解快捷方式，来更快地使用。



Doctor Workbench 说过...

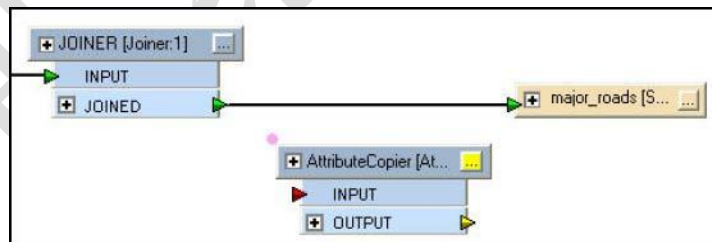
“如果你看到了一些粉红色的小点，不要担心！这并不表示出现了问题，而是 FME 的拖拽和插入功能，以便更快地放置函数！”

右图：在这里，用户将函数 AttributeCopier 插入到管道中。



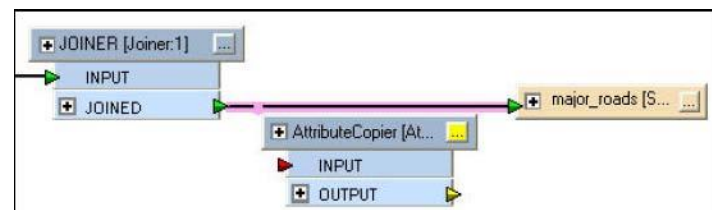
当你开始拖动一个函数时，这项功能就会自动开始运行，并且会在函数的左手边出现一个粉红色的小点。

右图：现在用户开始拖动函数，然后就出现了粉红色的小点。



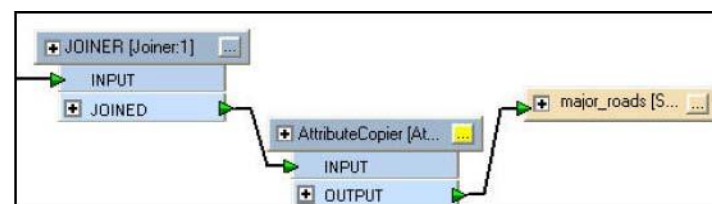
当将函数拖动到一个位置，粉红色的小点就会覆盖已有的连线，这样就会突出这条连线，这表示，你准备将函数插入这条线上的任何一个位置。

右图：用户在已有连线上拖动粉红色的小点，这条线就会自己变粗。



松开鼠标，函数就被插入了这个位置，并且自动进行连接

右图：用户将函数插入到了管道中。



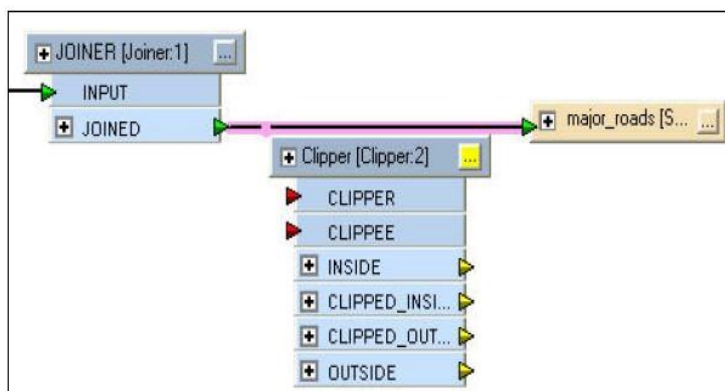
插入多个端口的函数

当你打算拖动和插入的函数只有一个单一的输入和输出端口时，FME 就会自动地将它们进行连接。

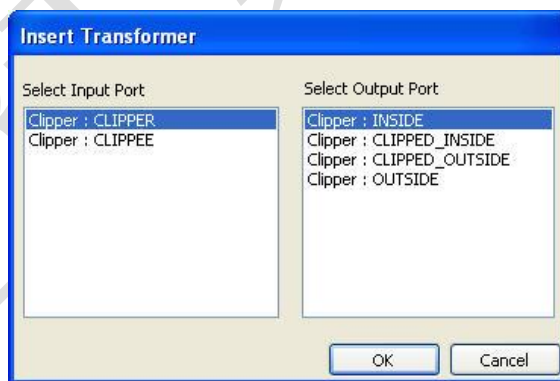
但是，许多函数不止有一个单一的输入或输出端口，所以就不能进行自动连接，这种情况下，按照提示，就需要用户在对话框中进行连接设置。

右图：在这里，要插入的函数是 Clipper。

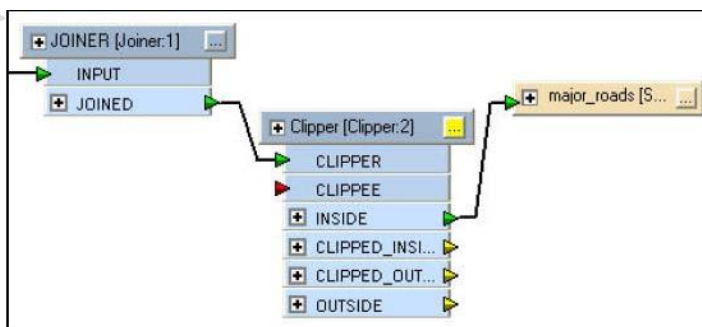
它有两个输入端口和四个输出端口。



右图：将 Clipper 函数拖到一个位置，FME 就会打开一个对话框，提示用户选择要连接的输入和输出端口。



右图：用户点击 OK 键，就会对选择的端口进行连接。



对要素类别进行拖动和连接



FME 2009 有一个新的功能，就是拖动，连接要素类别。通过这个功能，你就可以直接将要素类别附在函数上，而不是将函数附在要素类之上。

这样做并不是在大多数情况下都有特别的好处，但是许多用户已经习惯了使用 Drag-and-Insert 函数，同样地，他们也自动对要素类别尝试拖动和连接！现在，你就能使用它了！

基于组和基于要素的转换



FME 函数使用下面两种基本方法中一种对数据进行转换—基于要素或基于组。

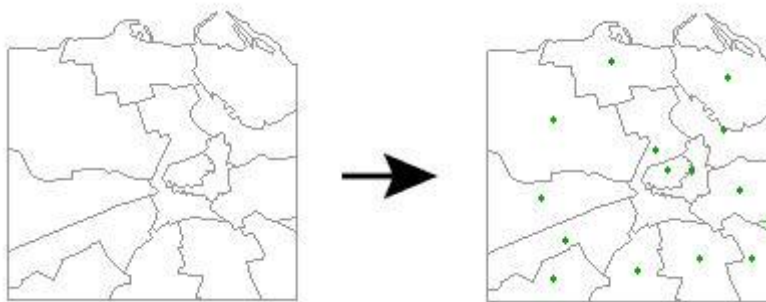
基于要素的转换

在这种转换方法中，一个函数在要素的基础上转换数据。一次只处理一个单一的要素，并且转换结果不会影响到其它要素的转换结果。

一般来说，相同数目的要素来自输入的函数。

下面是有关基于要素的转换例子...

- **Measurements** ——一次只计算一个要素的长度和面积，并且，不会影响到其它要素的长度和面积。
- **Line Generalization** ——依次缩编每个线性要素，并且不必参考周围的要素。
- **Centre of Gravity Calculations** ——FME 计算一个面积要素的“重心”（地理中心），并且，每个计算都是唯一的，不依赖于其他的要素。



上图：重心的计算



对于传统的算法，一些缩编功能在对一个要素进行处理时，会影响到其它的要素，但是，FME 则只会减少点的数目。

基于组的转换

对于这种转换方法，函数会处理一组要素。在进行转换之前，所有的要素就会被合并为一组，并且每个要素都会直接影响到其它要素的转换结果。

使用这种转换方法，转换后函数的要素数目很有可能不同于输入的要素数目。

下面是有关基于组的转换的一些例子...

- **Polygon Creation** – 当一组点组成一个封闭的形状时，FEM 就会创建一个多边形要素。线类要素就会合并，产生一个单一的面类要素。
- **Statistics Calculation –StatisticsCalculator** 为一组要素计算属性的平均值，最大以及最小值。
- **Intersections** – 计算一条线上的插入点一定要使用这种方法，这是因为每个输入要素都会影响到输出要素。



上图：插入点的处理



将要素概念/基于组的转换和几何/属性转换结合起来，就能创建一个矩形，来表示每种类型的转换。

	基础组	基础要素
几何要素	AreaBuiler	LineGeneralizer
属性	StatisticsCalculator	AreaCalculator

例如，AreaBuilder 就是一个有关基于组，几何转换的函数。

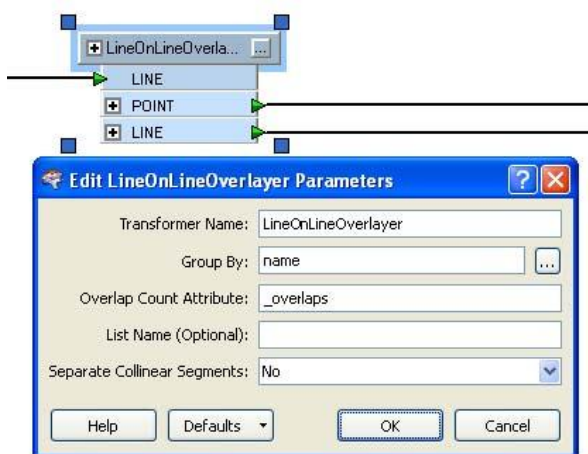
浏览 Workbench Transformer Gallery ，就可以找到其他的例子。

	基础组	基础要素
几何要素		
属性		

进行成组操作

默认地，一个基于组的函数会将所有经过它的要素编成一组，但是，许多情况下，要求有选择地将要素编成组。为了满足这一要求，许多函数都有一个选项—Group-By。

Group-By 选项允许对数据的属性进行选择。在开始转换之前，首先将相同属性值得要素归为一组。



左图：使用 LineOnLineOverlay 函数，将大量线性要素进行交叉。

选择“name”作为 Group-By 属性。

结果就是，每组中含有相同值“name”的要素被其他组所取代。

有用的输出就是，只有同名的线类要素才会相交形成交叉线。



例 4：进行成组操作

例 1, 2, 3 中已经计算了城市 Interopolis 中公园的平均面积。使用函数 AreaCalculator，平均面积为 1,507,766 平方尺。

但是，一些人指出，数据中的许多多边形都属于同一个公园—Big Walnut Creek，这些多边形应该被当做一个公园，而不是很多个，现在让我们来解决这个问题...

1) 启动 Workbench.

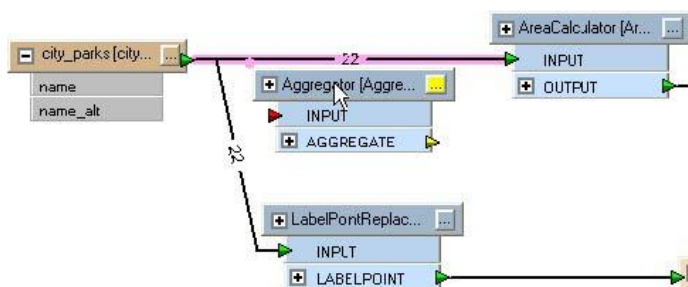
打开例 3 中的工作空间，或者重新创建一个。

2) 放置一个函数 – Aggregator

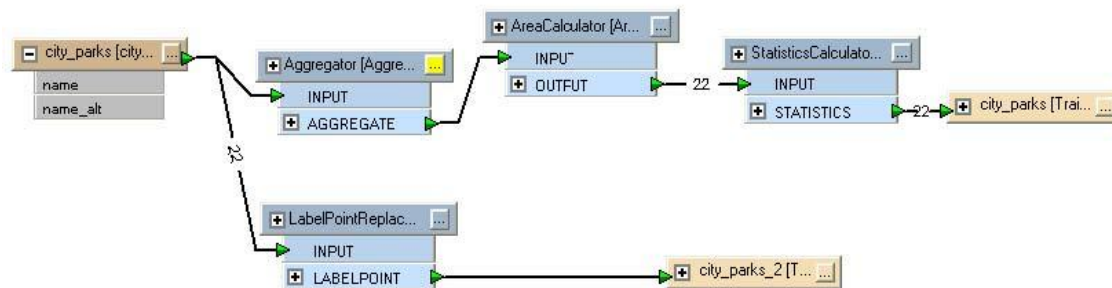
将函数 Aggregator 放置到工作空间中。

点击 Aggregator 函数，拖动它，直到记号位于源数据和 AreaCalculator 之间的连线上，这是，连线就会变成粉红色（如右图）：

放开鼠标，这时，就将函数插入到工作流中了。



下图：工作空间看上去就是这样的。



3) 检查函数的参数

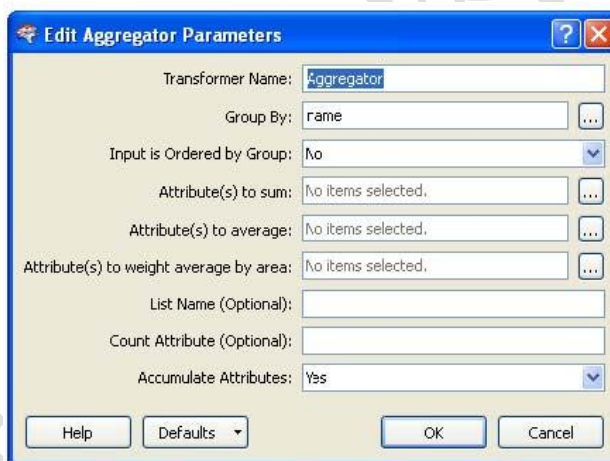
打开 **Aggregator** 设置对话框，将 **group-by** 属性名改为“name”（如右图）...

4) 运行工作空间

保存，并且运行工作空间。

如果公园的数量更少，但是总面积相同，平均值(**_mean**)就应该更大。

之前的值为 1507766 平方英尺，那么现在呢？



5) 测试问题

OK – 现在是时候来检查你学到了什么！

为什么我们使用 **Aggregator** 函数，而不是 **StatisticsCalculator** 中的成组选项呢？



InteropGeek68 说过...

“一些基于组的函数能够改善性能，而函数 **Aggregator** 就是其中的一个。

如果你已经知道数据是被归为一组的，那么就可以将“Input is Ordered by Group” 设置为” YES”，来查看其表现。

Clipper 和 **NeighborFinder** 函数也有相同的设置!”

坐标系转换



为了在地球表面将空间数据定位，大多数的空间数据都和特定的坐标系相关。

坐标系基础

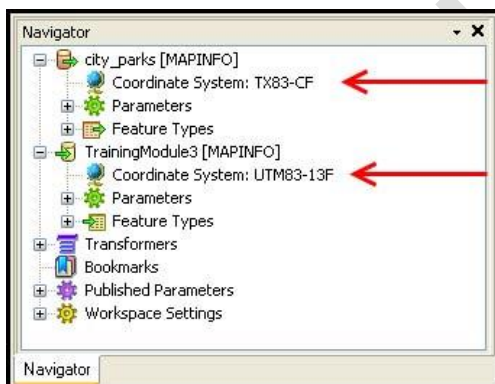
FME 处理的每个要素，实际上都有坐标系，例如，FME 就能随时知道要素是属于哪个坐标系的。当我们读取属于不同坐标系的多个数据集时，就不会感到困惑了。



一些用户将数据的这种位置叫做“投影”，事实上，它只是坐标系的一部分。一个准确的坐标系定义包括投影，大地水准面，参考椭球体，单位，有时还包括象限。

坐标系设置

FME 中的每个源读模块和写模块都有一个坐标系，这个坐标系被设置在 Workbench 的导航方框中



左图：这里，源坐标系和目标坐标系分别被定义为 TX83-CF，UTM8313F。

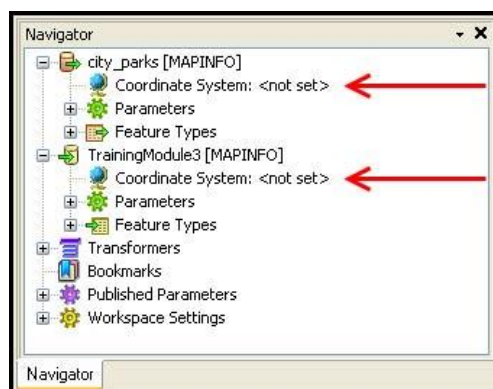
当进行转换时，以不同的方式处理源坐标系和目标坐标系，在转换过程的结尾，FME 会对数据进行结构重构，这样就能将输出结果放入正确的位置

坐标系的自动检测

一些数据本身包含它的坐标系的相关信息，形状格式就是一个例子。可以对 FME 进行设置，这样就能自动地检测任何相关的信息了。

右图：因为源坐标系设置成<not set>，所以 FME 会试着从源数据集中寻找坐标系

因为目标坐标系设置成<not set>，所以 FME 不会重新投影这些数据，但是会使用与源数据相同的坐标系对它进行编写。





例 5：进行今本的重新投影

例 1, 2, 3, 4 已经计算了公园的平均面积，但是，现在策划组要求将输出数据重新投影到一个不同的坐标系—UTM83-14F(Universal Transverse Mercator, Zone 14)，而不是 TX83-CF。

1) 启动 Workbench.

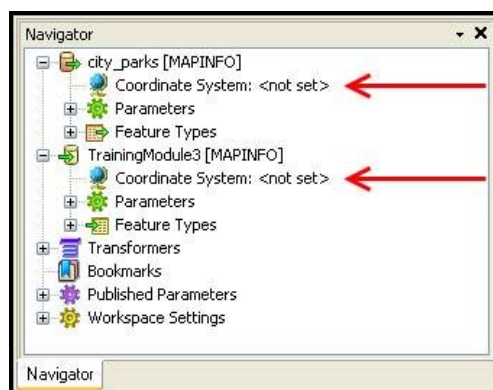
打开例 4 中的工作空间，或是重新创建一个。

2) 编辑源坐标系

在 Workbench 的导航方框中找到源公园读模块，然后单击[+]，展开设置列表。

找到标有“Coordinate System”.的设置，最开始的值必须设为“<not set>”(如右图)...

双击源坐标系设置，就会产生一个编辑参数对话框



输入坐标系名 TX83-CF，或者使用[...]按钮从坐标系列表中选择它。



3)编辑目标坐标系

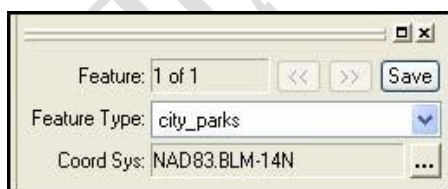
现在找到目标数据集的坐标系设置，同样地，将它的值设为“<not set>”。

双击它，输入坐标系名 UTM83-14F，或者使用[...]按钮从坐标系列表中选择它。

4) 开始转换

在日志文件中，你就能找到它，然后开始转换...

FME Configuration: Source coordinate system set to `TX83-CF' FME Configuration: Destination coordinate system set to `UTM83-14F'



5) 检查数据输出

检查输出数据，查询一个要素，就应该显示出新的坐标系（如左图）。

备注：坐标系名应该是 NAD83.BLM-14N，这是因为它与 UTM83-14F 具有相同的含义。

单元复习



这单元向你简单的介绍了数据转换的含义，以及教你如何使用 FME Workbench，而不仅仅是用它来进行快速转换。

从这单元中你应该学到什么？

以下就是你应该学到的内容：

理论

- 模式指的就是一个数据集的结构，包括，它的要素类别，属性，以及几何特征。将源数据与目标数据进行连接，就叫做模式映射。
- 数据转换指的是在转换过程中对数据进行结构重构。在 FME Workbench 中，使用函数就能进行转换，我们可以在函数列表中找到这些函数。
- 我们可以讲转换过程分为两类：基于要素或基于组。使用函数设置对话框中的 **group-by** 选项就能进行分组。
- 将数据分拆为多个数据流，会产生多个数据副本，但是不会改变数据。而将多个数据流进行整合是连接数据，而不是合并数据。

FME 功能

- 通过调整模式映射，能够对数据进行结构重构，通过手动设置或者函数都能完成。
- 能够在 Workbench 中找到函数，然后将它们放入工作流中（使用各种方法），然后转换数据内容。
- 使用组选项，定义组要素。
- 使用 Workbench 来重新投影数据。

回答问题

Mr Flibble's 面临的问题:

最常见得 FME 转换就是将 ESRI Shape 转换为。。。也就是说, 用户使用 FME 来进行转换, 而不愿那个转换器!



Miss Vector 说过...

“下面是问题的答案, 最好一个都不要错!”

“ ETL ” 中的 “ T ” 代表:

- 1) Translation
- 2) Transfiguration
- 3) Transmission
- 4) Transformation

✓

一个模式的源和目标部分表示的是:

- 1) What we have/What we want
- 2) What we did/What we should have done
- 3) What we're adding/What we're removing
- 4) Where my data was/Where it is now

✓

将源模式连接到目标模式, 这个过程叫做:

- 1) Schema Mapping
- 2) Schema Connecting
- 3) Schema Joins
- 4) Schema Concatenating

✓

下面的哪一种颜色, 不表示 FME 模式正在已经建立了连接?

- 1) Red
- 2) Green
- 3) Blue
- 4) Yellow

✓



将要素概念/基于组的转换和几何/属性转换结合起来，就能创建一个矩形，来表示每种类型的转换。

	基础组	基础要素
几何要素	AreaBuiler	LineGeneralizer
属性	StatisticsCalculator	AreaCalculator

例如，AreaBuilder 就是一个有关基于组，几何转换的函数。

浏览 [Workbench Transformer Gallery](#)，就可以找到其他的例子。

	基础组	基础要素
几何要素		
属性		

回答—有许多连接函数，可以添加到你的表格中。如果你对一些特殊的函数不确定，就问下你的指导员。记住，不是所有的函数都用来对数据进行结构重构的。