

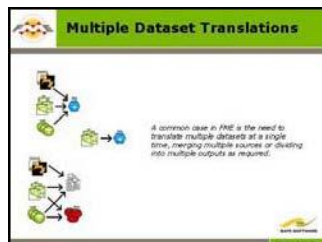
## 目录

多数据集转换.....	3
数据集和要素类.....	3
数据和模式.....	3
多源数据集.....	4
多重源转换.....	4
选择多个数据源.....	5
数据源浏览工具.....	5
多个数据源选择工具.....	6
工作空间 Navigation 面板 .....	7
编辑已有的工作空间.....	8
不同属性的数据集.....	8
不同要素类的数据集.....	8
The Unexpected Input Remover .....	10
处理新的源要素类.....	11
Import Feature Type Definitions .....	11
合并要素类.....	13
多种格式转换.....	15
多个读模块.....	16
添加 Reader .....	16
多个目标数据集的转换.....	18
多个写模块.....	19
添加写模块.....	19
Import Feature Type Definitions .....	19
添加 Feature Type Definition.....	19
对要素类的处理.....	20
编辑目标要素类.....	20
移动要素类.....	21
更新要素类.....	23
复制属性定义.....	24
模式映射函数.....	26
Filter 函数 .....	26



基本函数.....	27
Replace with AttributeCopier.....	28
Workbench 模式映射功能.....	29
要素连接方框.....	29
属性连接方框.....	30
单元复习.....	31
你应该从这单元中学到些什么? .....	31
疑难解答.....	32

## 多数据集转换



这个课程的前面内容仅仅讲解了对一个源数据集进行转换，但是，你经常会碰到一些情况，需要转换多个数据集。

大多数用户会使用 **FME** 来同时转换多个数据集；可能将多个源数据整合成一个数据集，也可能将一个源数据集分成多个数据集，或者是既整合又分离。**FME** 能够很灵活地按照需要将输入和输出数据集结合起来。

这部分列出了 **FME** 的一些工具，使用它们就能转换多个数据集。

### 数据集和要素类

这部分中，你需要准确地理解术语“数据集”和“要素类”之间的区别，以及“reader（读模块）”和“writer（写模块）”的含义。在学习下面的内容之前，复习一下这些概念，如下：

#### 数据集

它指的是经过有序组织的数据集合，并且能够在其中进行数据转换，例如，AutoCAD DXF file, an Oracle database, an ESRI Geodatabase。

#### 要素类

它指的是数据集内一个可辨别的子集，例如，AutoCAD file 中一个层，Oracle database 内的一个格，ESRI Geodatabase 的一个要素类。

### 数据和模式

这部分中，我们同样要提到模式的概念，模式是对一个数据集结构的正式定义。你可能对 **Data Model** 这个术语更加熟悉。

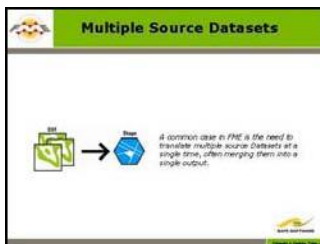
在这节中，我们将模式理解为一个实体，它与数据集中数据存在的方式相同。一个重要的不同点就是，它是完全独立的，不受数据集内容或数据格式的限定。



我们会发现，这样理解模式，就能够更简单地理解 **FME** 是怎样从任意数据集和任意格式来抽取模式，并且使用它来重新定义目标数据集；或者如何在 **Workbench** 中移动一个模式定义，多次在工作空间中复制这个模式。

Mr DonutBuilder 说过...

“就像一个面包师，大多数 **FME** 用户需要批量操作，数据集和要素类就是重要的原料，而将它们融合在一起的功能指的就是，使用 **FME** 来进行转换。”

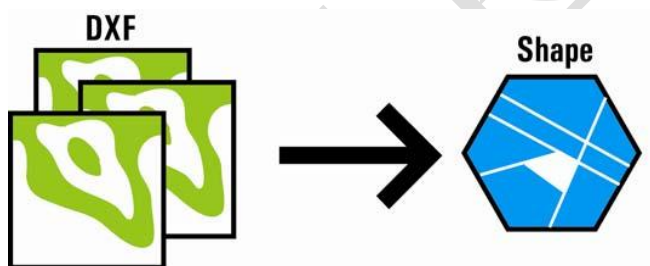


## 多源数据集

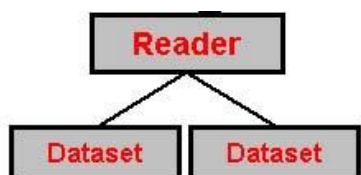
处理多源数据集最简单的方法就是将它们整合成一个单一的输出数据集。

### 多重源转换

下面我们举的第一个例子是最简单的，转换多个源数据集。要记住，读模块能够读取多个数据集，而写模块通常只能处理一个单一的数据集，所以你要做的就是将多个源数据集整合成一个目标数据集。



右图：这个图表示基本的多重源形状。



用户拥有大量的 AutoCAD DXF 文件，希望将它们转换成一个 ESRI Shape 的数据集。

在创建一个工作空间，或是编辑一个已有的工作空间的过程中，用户可以使用 FME 中的大量工具来选择多个源数据集。



左图：当你用这种方法创建或调整一个工作空间时，事实上，在这个工作空间中，只有一个读模块，但是可以使用它来导入多个数据集。

将这个概念理解为“数据源”可能更加简单，因为使用它能够选择数据集，或者要素类，而“文件”并不是那么的准确，因为并非所有的数据集都是以文件的形式进行保存的。

## 选择多个数据源

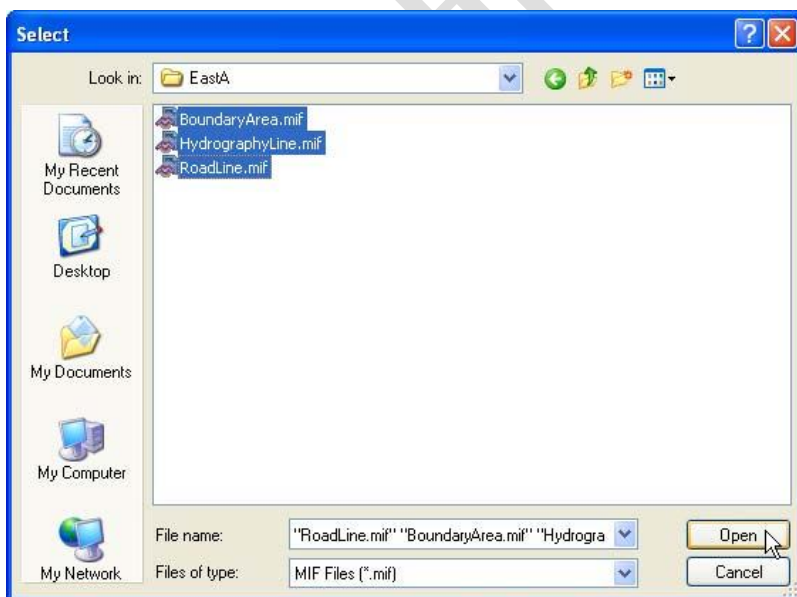
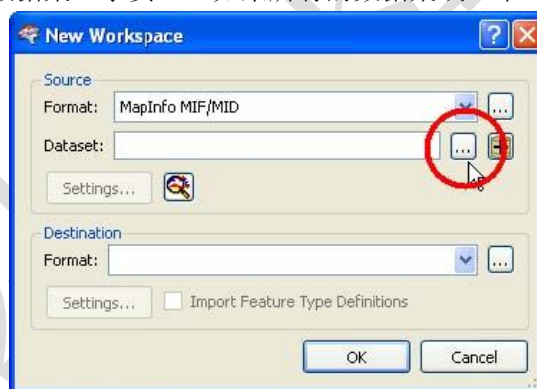


Workbench 有大量的工具来选择数据，每个工具都能够选择单一或多个数据源，但是它们运行的方式不同。

### 数据源浏览工具

我们之前也用到过这个工具，但只是用来选择一个数据集，事实上，如果所有的数据集或一个单一数据集的所有要素类都保存在同一个位置时，也可以使用它来选择多个数据源。

右图：在 New Workspace 对话框中，点击浏览按钮，就可以访问这个工具了。



左图：数据集浏览器界面。

已经选择了‘EastA’数据集的所有要素类。

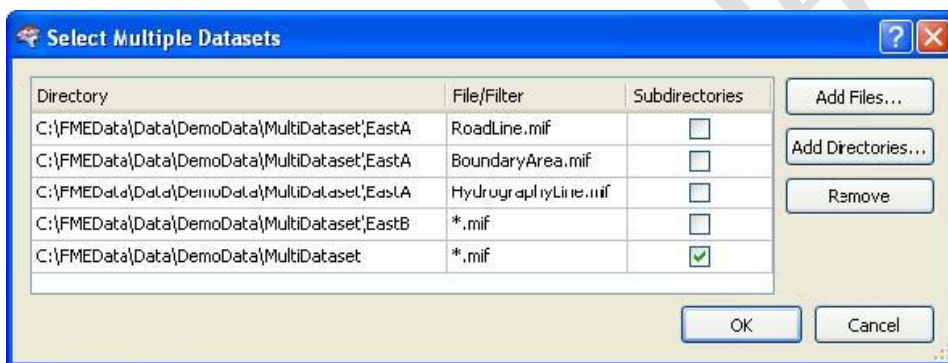
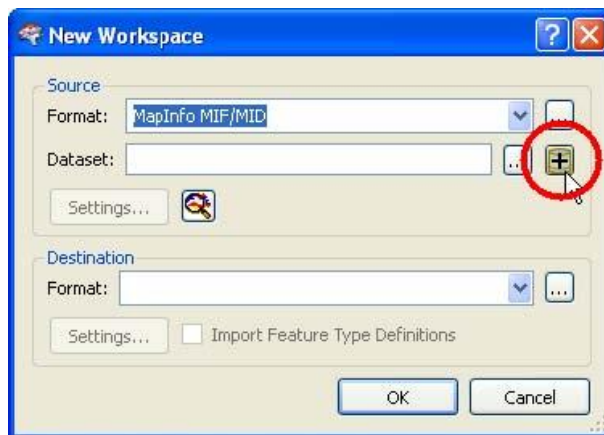
这个例子显示了选择多个要素类，而不是多个数据集。

当将数据平铺时，通常都会以这种方式进行结构组合，因此我们也可以假设 EastA 就是一个结构名。

## 多个数据源选择工具

当数据源分散在不同的位置时，就必须使用 multiple Datasource Selection Tool 来进行选择，通常，这个工具也被叫做‘Swizzler’。

右图：在 New Workspace 对话框中点击 Add source 按钮，就可以访问这个工具了。



上图：数据集选择工具界面

在上面的例子中，使用 “Add Files”按钮来单独选择 EastA 中的三个要素类。

然后，使用‘Add Folder...’ 按钮就能够添加另一个要素类，就是用这种方式来选择 EastB 文件夹的。

但是，从不同的文件夹中选择大量数据源的最简单方法就是，选择一个文件夹（例如，C:\FMEData\Data\DemoData\MultiDataset），然后选中‘subfolders’，确保它包含了所有的数据集或要素类。

注意，swizzler 只能够选择一种格式的数据。



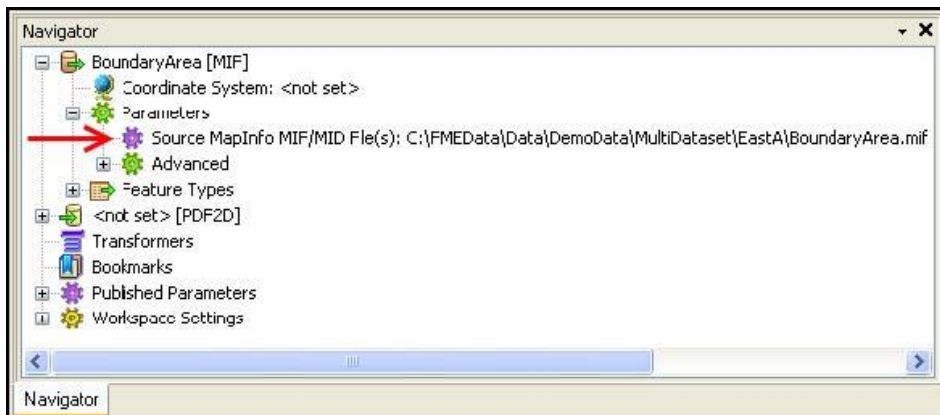
一般来说，使用以上方法中的任何一种来选择多个数据源，都需要创建一个工作空间，这个工作空间中的所有源数据都能发送到一个输出数据集。

将每个源数据集转化为不同的目标数据集，就要求使用一些高级功能，例如，Data Fanouts 或 Batch Processing。

## 工作空间 Navigation 面板

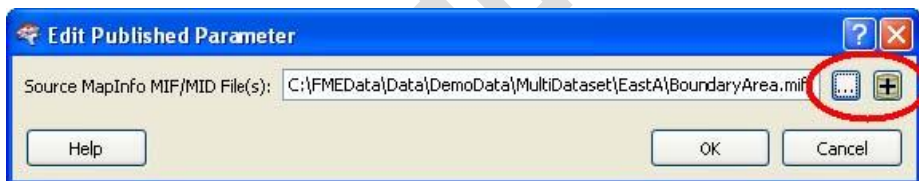
前面的例子展示了怎样创建一个新的工作空间，并且使用它来读取多个数据源，但是，要选择多个数据源并不局限于通过创建工作空间来实现。

Workbench 也能够已在有工作空间的导航方框中选择多个数据源。



上图：源数据的导航参数显示了 MIF/MID 数据集的源数据定义，双击它，然后打开一个选择对话框。

下图：可以在这个对话框中启动数据源浏览和选择工具，就可以使用它们来选择数据集或要素类。



你是否注意到了，上面出现的源数据参数有一个紫色的图标？这时因为 FME 已经默认地发布了所有的源参数和目标参数。

所以，在运行的时候仍可以使用 Prompt + Run 来选择数据源，并且不需要在导航方框中预先设置这个选项。



## 编辑已有的工作空间



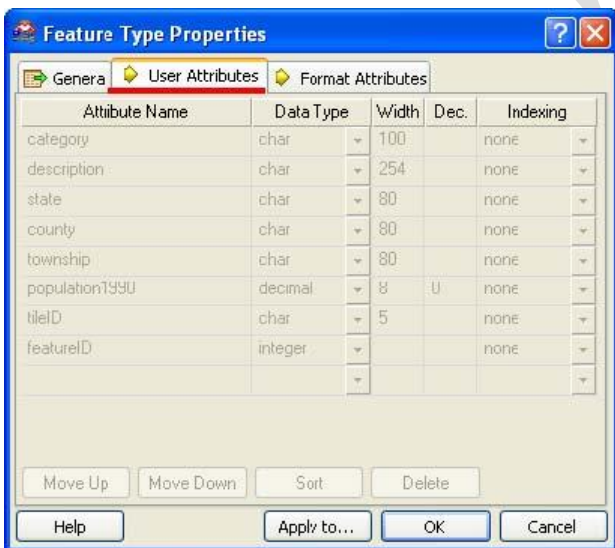
在一个已有的工作空间中很轻易地就能改变源数据集，可能是因为它太简单了，所以在操作时就可能会产生许多潜在的问题。

正如前面所说的，能够很轻易地改变已有工作空间的数据源，但是并不能保证这些数据源在随后的转换中能成功，除非是满足了某些条件。

在开始创建工作空间时，FME 就已经为源数据集创建了一个模式，当这个源数据集被取代了，而新数据集又不符合之前的模式，这时就会出现问题了。

### 不同属性的数据集

前面已经提到了，一个模式包括数据集的属性定义。如果改变一个源数据集，而新的数据集含有完全不同的属性，那么可能就不能正确地读取这个新数据集，也不能将它的属性编写到目标数据集中。



左图：一个 MIF 数据集的 User attributes。

如果改变源设置来读取含有不同属性的要素类，原来的列表并不能够自动更新列表。

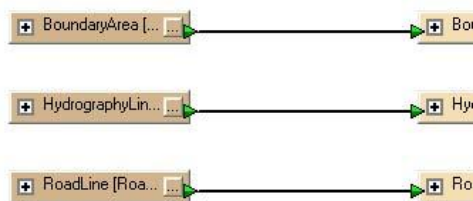
FME 仍然认为要处理原有的属性，但是接收到的却是不同的内容。

### 不同要素类的数据集

前面也提到过，一个模式包括数据集要素类的定义。

如果改变一个源数据集，而新的数据集含有完全不同的要素类，那么 FME 并不能正确地读取这个新数据集，并且新要素类中的数据(还没有被定义)也会被丢弃。

右图：这个数据集含有大量的要素类。如果将源数据集改变成一个文件，这个文件含有不同的层，就能够读取这些层，但是之后会被过滤掉。







### Example 1: 一个不成功的转换

Interopolis 城市打算进行一个叫做 **Official Community Plan** 的基本规划。**CommunityPlanBasemap2008.dgn** 就是 2008 年计划中要使用到的数据集, 它将作为底图使用。

规划部门要求你将 **Community Plan** 数据集转换为 **Adobe Geospatial PDF** 格式的数据集。

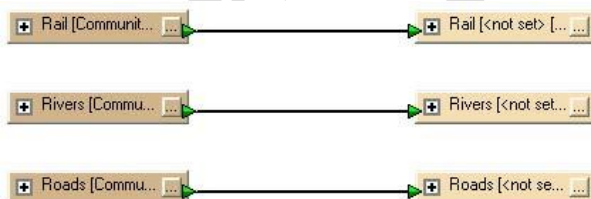
#### 1) 启动 Workbench

如有需要就启动 **Workbench**, 使用下面的参数来创建一个工作空间:

Source Format	Bentley MicroStation Design (V7)
Source Dataset	C:\FMEData\Data\CommunityPlan\CommunityPlanMap2008.dgn
Destination Format	Adobe Geospatial PDF

注意: 一定要点击 'create workspace' 对话框中的 **source settings**, 并且将 **Group Elements By** 设置成 'Level Name'。

观察这个新的工作空间是怎样显示马路, 河流以及铁路的要素类的。



#### 2) 更改源数据集

在创建了工作空间后, 规划部门来电说他们已经有了一个新的 2009 版本, 希望能用它取代旧的 2008 版本。

New Source Dataset C:\FMEData\Data\CommunityPlan\CommunityPlanMap2009.dgn

在导航窗口中, 更改源数据集参数, 就会出现新的数据集—**CommunityPlanBasemap2009.dgn**。保存这个工作空间, 然后开始转换



这个转换就是成功的, 因为它准确地从源数据中读取了要素, 并编辑到目标数据中。

但是, 对话框和日志窗口都会显示之前读取了要素类 (但又被滤掉了), 这就是说, 2009 数据集含有 2008 数据集没有的要素类。

这个例子说明了, 即使只是更新原来的数据集, 但是, 用来处理一个数据集的工作空间可能并不适合其它的数据集, 就算它们的格式相同。

## The Unexpected Input Remover

每次 FME 读取一个数据集，它就会在数据集中检查要素类，以确保在工作空间模式中这些要素都被定义了。如果一些要素类在数据集中，但是却不在工作空间中，这时，这些要素就被定义为未知的，并且会被过滤掉，这种功能就叫做 Unexpected Input Remover。

```
[STATS |Unexpected Input Remover(TestFactory): Tested 4 input features -- 0 features passed, 4 features failed.]
```

Unexpected Input Remover 的操作会在转换过程最后会出现一个对话框，并且在日志文件中会报告结果。这个操作一。

右图：这里，因为未知的输入数据，FME 已经丢掉了例 1 转换过程中出现的大量要素。

注意每个受影响的要素类都被列出来了，包括受影响的要素数量。

另外要注意选中下面的复选框可以在以后的转换过程中关闭该对话框。



当使用文件夹形式的数据格式时，一些用户会感到很困惑，因为，他们认为所选择的源文件是一个数据集，而事实上，却是一个要素类。

因为我们将每个源文件的名称当做要素类名来使用，所以，更改源文件名能够确保激活 Unexpected Input Remover 功能，这是因为要素类不再匹配这个工作空间了。

## 处理新的源要素类



使用 Unexpected Input Remover 带来的问题是就会丢失掉一些要素类。

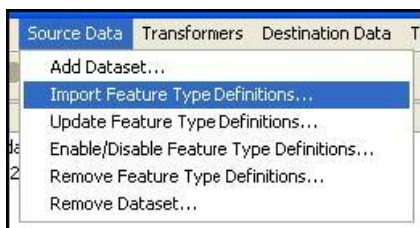
Workbench 提供了两种方法，解决 Unexpected Input Remover 产生的问题。

第一种也是最常见的方法—将丢失掉的要素类添加到工作空间。

第二种方法—允许传送哪些未定义的要素类。

### Import Feature Type Definitions

Import Feature Type Definitions 是添加要素类到工作空间的一个工具，通过它能够添加遗留的要素类，并且传送这些数据。



左图：从菜单条中就能选择 Import (source) Feature Type Definitions 工具。

通过输入任何一个数据集的模式就能够使用 Import Feature Type Definitions 工具，添加要素类到一个已有的数据集中。第二个数据集可能与读取的数据集格式完全相同，但是这并不是必须的，事实上，可以是不同的格式，这是因为模式定义是一个很重要的内容，而 FME 可以读取任何格式的模式。



### Example 2: 修补错误的转换

例 1 中，创建一个工作空间来转换 Community Plan 所需要 2008 基本图的数据，但是这个工作空间并不适用于转换 2009 基本图的数据。该工作空间不能完成转换，因为缺少新的要素类（已经添加到了 2009 年的更新文件中），下面这个例子就会解决这个问题：

#### 1) 启动 Workbench.

如果有需要就启动 Workbench，打开例 1 的工作空间，或是重新创建一个，你也可以在 C:\FMEData\Workspaces\TrainingWorkspaces\S5 - Example 2 - Begin.fmw 中找到一个副本。

#### 2) 添加要素类

在菜单条中选择 Source Data > Import Feature Type Definitions，然后设置数据集：

Source Format	Bentley MicroStation Design (V7)
Source Dataset	C:\FMEData\Data\CommunityPlan\CommunityPlanMap2009.dgn

然后就会出现一个对话框，从中选择要输入的要素类，接着，你要做的就非常简单了，只需要点击 OK，FME 就会自动判断工作空间中漏掉了哪些要素类，然后会自动添加这些要素类。

备注：一定要将“Import”对话框中的 setting 设置成 Group Elements by Level Name。

#### 3) 添加要素类

上个步骤中已经修复了源数据集的问题，现在轮到目标数据集了。在菜单条中选择 Destination Data > Import Feature Type Definitions，将数据集设置如下：

Source Format	Bentley MicroStation Design (V7)
Source Dataset	C:\FMEData\Data\CommunityPlan\CommunityPlanMap2009.dgn

然后就会出现一个对话框，从中选择要输入的要素类，然后，你就需要选择要特别添加的要素类，因为如果你直接简单地点击 OK 键的话，FME 就会选择要添加所有的要素类。

重新选择要素类，仅仅选择有关 Buildings 的要素类，然后点击 OK。

注意，已经添加了 Building 要素类到已有的目标数据集中，然后就要将源 Building 要素类和目标 Building 要素类连接起来了。

#### 4) 添加要素类

我们会采取不同的方法来处理目标 Schools 要素类，右击源 Schools 要素类，选择“Duplicate (as Destination)”。

现在就已经添加了 Schools 要素类到已有的目标数据集中，然后将源 Schools 要素类和目标 Schools 要素类连接起来。

#### 5) 重新开始转换

重新保存工作空间，然后重新开始转换。检查日志，确保已经转换了所有的源要素，现在，你可以发送数据给规划部门了。

这个例子展示了，怎样修改已创建的工作空间（用来处理一个数据集），使它也能够处理其他相同格式的数据集。



Professor Spatial 说过...

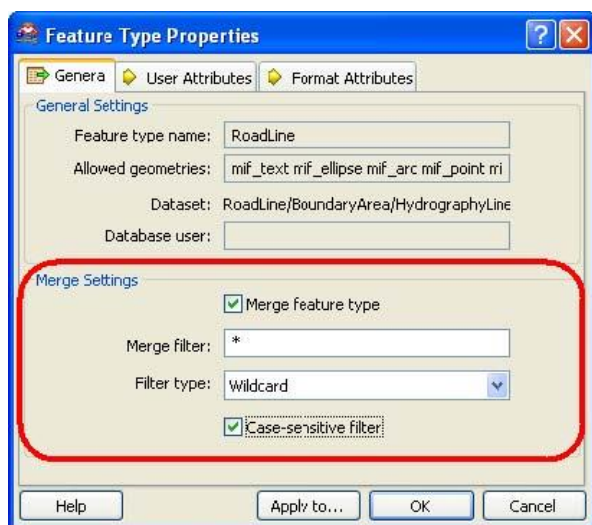
“将一个模式想象为一个对象，但是并不局限于一种特定的格式。所有说，我们有可能从一个数据集中引入一个模式，再导入到完全不同格式的读模块或写模块中。

## 合并要素类

解决 Unexpected Input Remover 所产生的问题的第二种方法就是，引导 Workbench 处理要素类，而不管这些要素类是已知的还是未知的，这样就能够添加新的要素类了。

合并要素类是一个工具，打开已有要素类的 properties 对话框，选中 Merge feature type 复选框，就能够设置这个工具了。Merge Feature Type filter 决定允许通过的要素类，过滤类型可以是一个通配符，也可以是常规表达。

在进行转换时，会将所有未知的要素类与这个过滤器进行比较，从而检查它们的名字是否是匹配的。如果是匹配的，就会在工作空间中允许这个要素类通过。



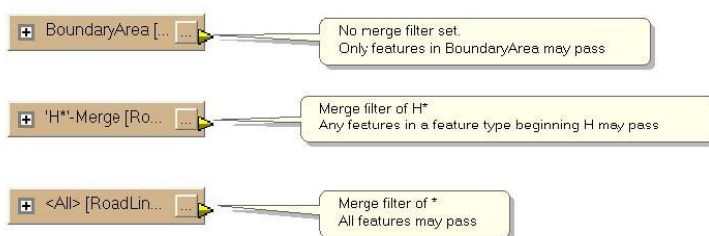
左图：这个例子中，用户已经在‘Merge feature type’这一栏中打钩，并且输入了一个通配符。

这就意味着，虽然要素类的名字是‘RoadLine’，但是其它要素类，例如，‘HydrographyLine’，‘BoundaryArea’，也能够通过这个要素类。

如果将 filter 设置为“R”（例如，以字母 R 开头），那么也能够通过名为‘RoadLine’的要素类，但是其它要素类（例如，‘BoundaryArea’），就不能通过了。



在 Workbench 中，检查工作空间中的对要素类的描述，就能够决定对哪些要素进行合并过滤，例如，‘<All>’就意味着已经设置了合并过滤。。。





Miss Vector 说过...

“大家注意了！现在又是时候来做一个小测验了，检查大家学到的内容。询问其他学员，然后回答这些问题。”

输入要素类定义和合并过滤解决了哪些问题？

- 1) 读取含有不同属性的数据集
- 2) 读取含有不同要素类的数据集
- 3) 读取含有不同格式的数据集
- 4) 读取含有不同几何类型的数据集


输入要素类定义主要是进行什么操作？

- 1) 添加丢失的要素类
- 2) 允许任何要素类通过
- 3) 终止 Unexpected Input Remover
- 4) 更新已有要素类的属性


一个合并过滤执行什么任务？

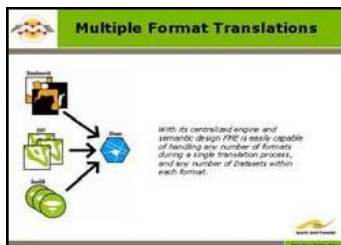
- 1) 添加丢失的要素类
- 2) 允许任何要素类通过
- 3) 终止 Unexpected Input Remover
- 4) 更新已有要素类的属性


你怎么看待输入要素定义和合并过滤这两种方法？例如，它们各自的优点和缺点分别是什么？

与其它同学讨论这个问题，然后将你们的观点写下来。



## 多种格式转换



这个课程前面仅仅介绍了对一种格式的数据进行转换，即使是在读取多个源数据集的时候，但是，你经常会碰到一些情况，需要从多种源格式中转换数据。

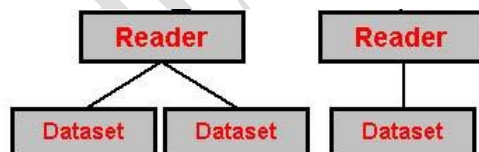
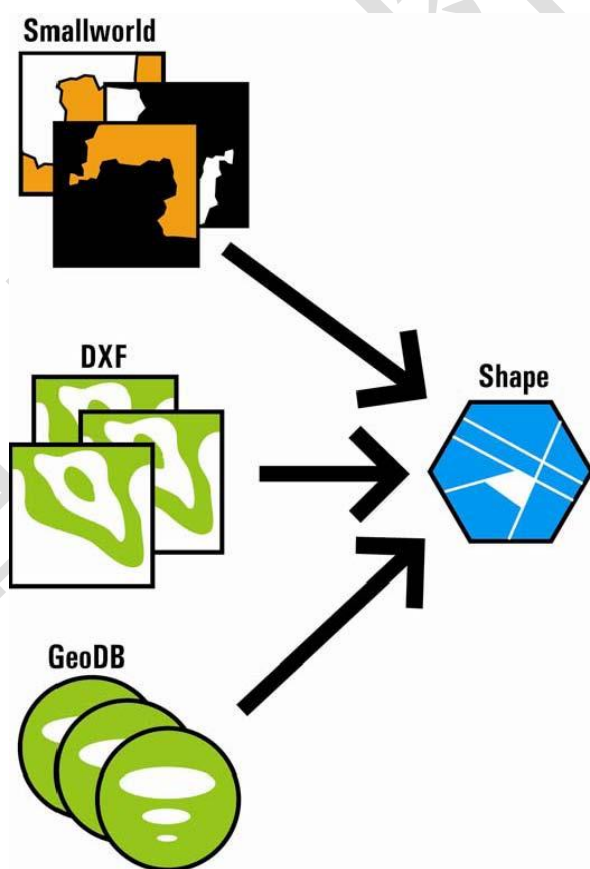
一个有关多个源数据集的转换不会假设每个数据集的格式都相同。幸运的是，FME 的集中化引擎和语义设计能够轻易地处理单独转换过程中的任何格式，并且处理每种格式中的任意数量的数据集。

右图：这个图表表示的是一个多种源格式的转换。

将大量的 AutoCAD DXF files, Smallworld Objects 和 GeoDatabase Feature Classes 合并为一个单一的 ESRI Shape 数据集。

下图：当你用这种方式创建一个工作空间时，事实上你创建了这么一种工作状态：存在多个读模块，并且每一个读模块都能读取多个数据集。

这些读模块可能是不同格式的，也可能是同一种格式，这就是为什么这章介绍的内容仍然能应用到在属性模式互不相同的情况下，读模块仍然能读取相同格式的多个数据集的情形。





## 多个读模块



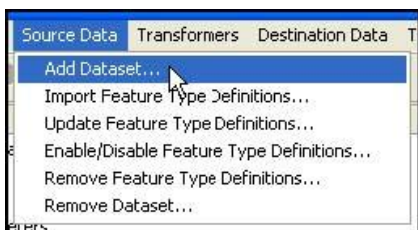
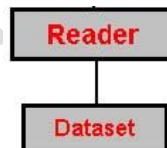
在已有的工作空间中添加读模块，就能够让 FME 进行多种格式的转换。

### 添加 Reader

因为工作空间创建对话框只能够创建一个单一数据格式的工作空间，在创建工作空间后，仍然有需要向其添加不同格式的数据集。

使用 ‘Add Dataset’ 工具，添加一组新的数据到工作空间中。

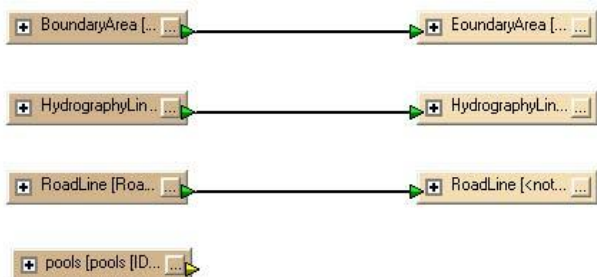
备注：菜单条说 “Dataset”，而我们说“Reader”。为了使术语更加准确，我们要做的就是创建一个新的读模块，并且在这个读模块中添加一个源数据集，所以两种说法都是正确的操作。



左图：在菜单条中选择 Add (source) Dataset 工具。

使用 Add Dataset 工具就能够添加一个读模块，一个数据集（包括所有的要素类）到工作空间中。这个读模块可能与已有的读模块格式相同，也可能不相同。

下图：这个例子中，使用含有三个要素类的源数据集来创建工作空间，然后添加一个源数据集不同格式的数据集（含有一种要素类）。





Doctor Workbench 说过...

“虽然这部分是有关多种格式的，但是使用 **Add Dataset** 工具在工作空间中添加相同格式的数据集，同样是非常有用的。

当我们需要在两个数据集中改变读模块设置时，单独地添加数据集就显得更加合适。”

### Example 3: 多个读模块

在例 2 中，已经创建了一个工作空间，为 **Community Plan** 转换 2009 底图。

规划部门非常满意你之前对数据的处理，但是他们希望在工作空间中添加更多的源数据，并且将它们编写到输出文件中：一个关于发展的建议书

备注：这是 FME 用户常见的一个问题；你对工作是如此的熟练，以至于经常会有增加新数据集或已新数据集的要求。很抱歉！

它们希望添加的数据是相同格式的（例如，**MicroStation Design**），但是，这可能出现一个问题。对这个数据集来说，它们希望你删除属性信息，但是又希望在 **Community Plan** 数据集中保存属性信息。

### 1) 启动 Workbench.

如果有需要就启动 **Workbench**，打开例 2 中的工作空间，或重新创建一个，你也可以在 **C:\FMEData\Workspaces\TrainingWorkspaces\S5 - Example 3 - Begin.fmw** 中找到一个副本。

### 2) 找到参数.

找到源 **MicroStation** 参数—“Strip off Tag Information”。

你认为这是一个 读模块级别参数，还是一个要素类级别参数？

也就是说，这个参数会影响所有已读取的数据，或仅仅是一个要素类？

如果它是一个要素类级别参数，你就可以添加新的数据集：

- a) 使用 **Source Data > Import Feature Type Definitions** 添加要素类
- b) **Adding the source** 通过编辑“Original Design File”参数，添加源数据集
- c) 在要素类的基础上改变设置

但是，如果它是一个：读模块级别参数，你就必须要添加数据集：

- a) 使用 **Source Data > Add Dataset**
- b) 在读模块级别的基础上改变设置

### 3) 添加数据集

使用正确的方法，添加有关发展计划的数据集，以下是数据集的细节：

Source Format	Bentley MicroStation Design (V8)
Source Dataset	C:\FMEData\Data\Developer\DeveloperChanges.dgn

### 4) 运行工作空间

在已有的 PDF 中添加（连接）一个符合的目标要素类，运行工作空间，确保已经正确读取了所有的数据。

## 多个目标数据集的转换



这个课程前面的内容仅仅介绍了多个源数据集（或要素类）的转换，但是，它的大部分功能同样适用于多个目标数据集的转换。

编写多个目标格式和数据集与读取多个源格式和数据集非常相似，所以，不难发现 FME 也能够对多个目标数据集进行转换，并且用来转换的工具与处理源数据集的工具非常相似。

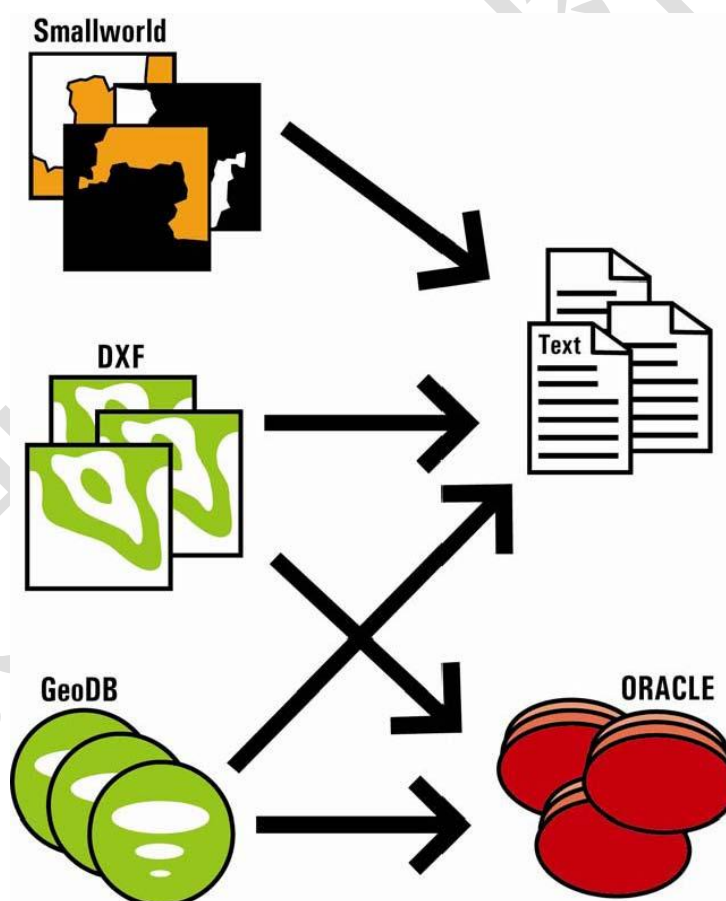
右图：这个图标显示，将多个源数据集和格式转换为多个目标数据集和格式。

读取大量的 AutoCAD DXF files, Smallworld Objects 和 GeoDatabase Feature Classes 将所有的源数据集合并为一个目标文本文件。

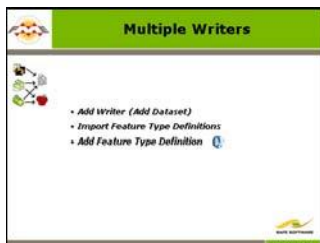
同时，将 DXF files 和 GeoDatabase 编写到单一的 Oracle 空间数据集中。

这个图标显示，FME 并不要求转换就一定是一对一的结构转换；可以将多个源格式和数据集连接到多个目标格式和数据集。

当你用这种方式创建一个工作空间时候，你也创建了一种工作状态，存在多个写模块，通常每个写模块只写入一个数据集（除非使用 fanout）。



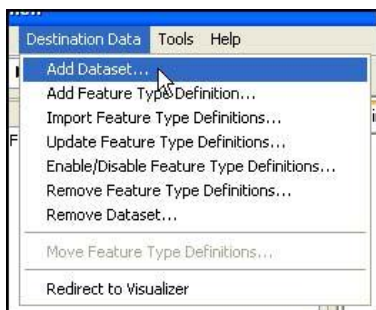
## 多个写模块



与源数据集一样，Workbench 拥有大量的工具来处理目标要素类

### 添加写模块

与源数据集一样，用户能够在工作空间中添加一个写模块。这里使用的指令就是 Destination Data > Add Dataset



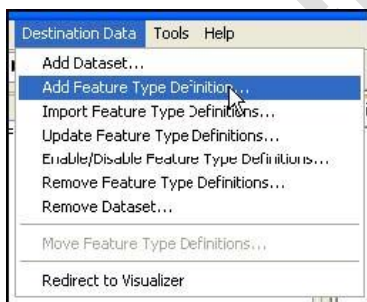
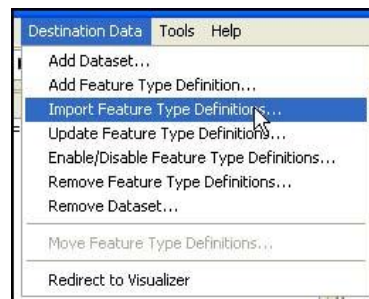
左图：从菜单条中选择 Add (destination) Dataset 工具。

注意：同样地，为了使术语更加准确，我们真正要做的就是创建一个新的写模块，并且添加一个目标数据集。

### Import Feature Type Definitions

与读模块相似，可以从已有的数据集中输入要素类定义，并且，提供要素类的数据集也不必要是相同的格式。

右图：从菜单条中选择 Import (destination) Feature Type Definitions.



### 添加 Feature Type Definition

与读模块不同的是，写模块能够手动地添加一个新的目标要素类。

左图：从菜单条中选择 Add Feature Type Definition 工具。



你为什么认为 FME 含有一个 Destination Data > Add Feature Type Definition，而不是 Source Data > Add Feature Type Definition 呢？

究竟是什么区别导致 FME 不允许源数据集使用 Add Feature Type Definition 工具呢？

## 对要素类的处理



为了给用户最大的自由，FME 提供了大量的方法，处理一个要素类的模式定义。

### 编辑目标要素类

正如你所知道的，目标要素类是对“我们要创建什么”的一个简单定义，而不表示（至少并不总是）已经存在的一些事物的表达。

基于以上的原因，用户完全可以以各种不同的方式来编辑一个目标要素类，然而用户却不能编辑源要素类。

使用 Feature Type Properties 对话框（如右图）就能够获取这些方法。



### 对目标要素类重新命名

点击 Feature Type name，输入一个新的名字，就能够编辑要素类名了。例如，在上面的操作中，选择或删除文本 ‘Roads’，然后输入 ‘Highways’，就能够对要素类重新命名了。



如果你记住要素类名“Roads”与“roads”或“ROADS”是不一样的，因为要素类名字是区分大小写字母的，就能够避免许多容易困惑的问题。

### 改变几何属性

当 FME 自动创建一个目标要素类时，会假定目标几何属性与源数据集的几何属性相同，例如，一个包含线类要素的源数据集就会产生相同的目标数据集，但是，并非总是如此。可能会使用工作空间中的一些函数，改变要素几何属性。

如果在转换过程中要素几何属性会发生改变，那么目标格式就会对允许的几何属性进行严格定义（Shape 数据集会要求这点），然后就能在列表中选择几何属性类别。虽然这并不是所希望的，但绝对是重要的，因为这样能避免在之后的转换中，由于几何属性的改变而导致错误。



## 移动要素类

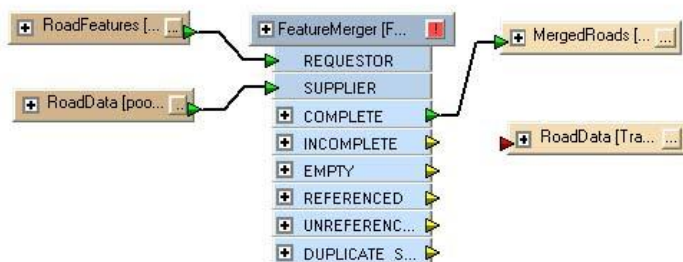
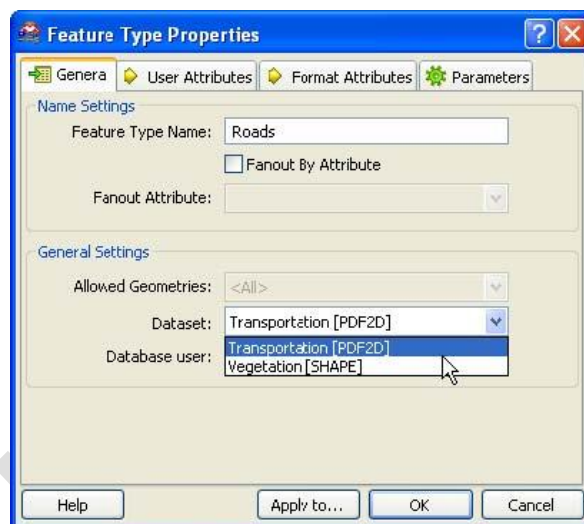
正如前面所提到的，可以将模式理解为一个独立的对象，使用它就能够处理任何目标数据集。按照要求，它能够允许要素类从一个数据集迁移到另一个数据集。

备注：仅从技术上说，我们在写模块之间，而不是在数据集之间进行切换，但是，因为每个写模块只有一个数据集，所以点是非常抽象化的。

无论何时出现多个目标数据集，目标 **Feature Type properties** 中的 **Dataset** 设置都会被激活。通过更改数据集，用户就能够有效地将要素类从一个数据集中转移到另一个数据集。

移动要素类并不是一个常见步骤，但是绝对能为你节省大量的时间。

Right: 在 **Vegetation** 数据集这一栏下，现在定义 “**Roads**” 要素类，这个图标中，用户正将这个要素类转移到 **Transportation** 数据集。

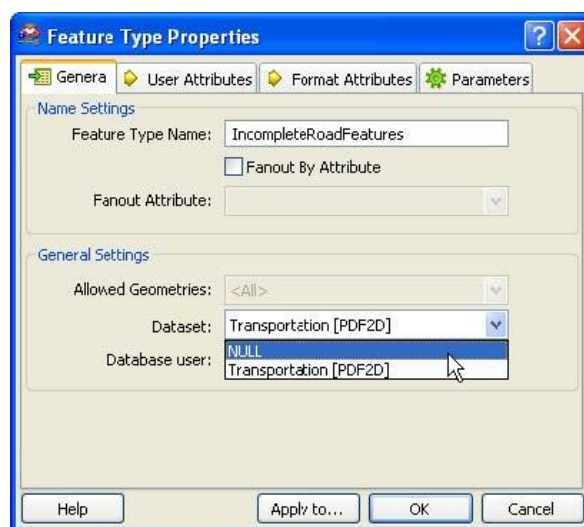


左图：这个例子显示了，为什么你要转移要素类。这里，用户已经创建了一个工作空间来转换马路数据，并且已经放置了一个 **FeatureMerger** 来将马路要素连接到一组非空间属性中。

右图：但是，现在用户希望记下不完整的记录，所以创建了一个 **NULL** 目标数据集，并且，他将剩下的 **RoadData** 要素类转移到了 **NULL** 数据集，而不是重新创建一个要素类，并且对这个要素类重新命名。



这个例子显示了，怎样在不同的数据集之间转移要素类，并且显示了模式是非常灵活的对象，并受限一个特定的格式。



**Example 4: 多个写模块**

规划部门，请注意了！有关交通的要素类是错误的，你应该在将要素类编辑到 PDF 之前，对它进行编辑。你能够将这些要素类编辑到 ESRI Shape 中吗？这样就能在 GIS 应用程序中编辑它们了。

为了完成这个操作，我们需要添加一个新的写模块到例 3 中的工作空间。

**1) 启动 Workbench.**

如果有需要就启动 Workbench，打开例 3 中的工作空间，或者重新创建一个，你也可以在 C:\FMEData\Workspaces\TrainingWorkspaces\S5 - Example 4 - Begin.fmw 中找到一个副本。

**2) 添加写模块**

使用 Destination Data > Add Dataset，添加一个新的写模块到工作空间中。

Destination Format ESRI Shape

Destination Dataset C:\FMEData\Output\TrainingModule5

当提示用户是否要添加一个新的要素类时，点击 NO。

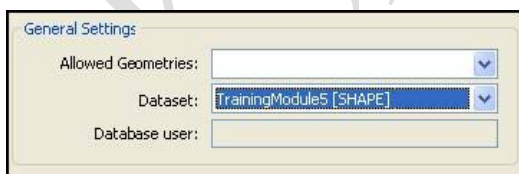
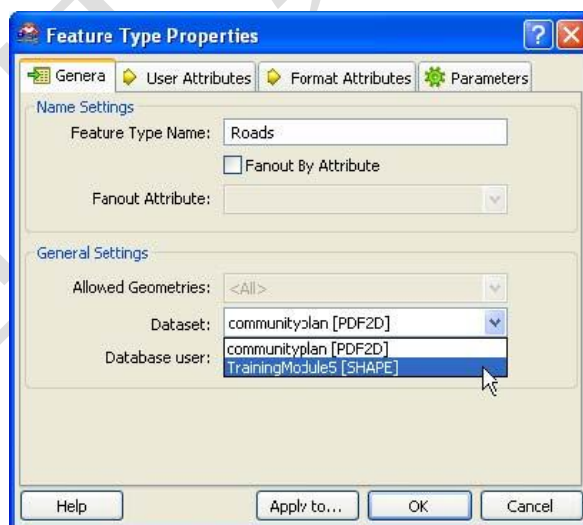
**3) 转移要素类**

打开有关目标要素类 “Roads” 的 Feature Type properties 对话框。

检查数据集参数，你会发现，因为现在有多  
个写模块，这个参数已经被激活了。

将参数 CommunityPlan [PDF2D] 改为  
TrainingModule5 [Shape] 。

通过以上操作，就将要素类从 Adobe 2D PDF  
转移到了 Shape 写模块中

**4) 改变参数 Allowed Geometries**

注意，一旦转移了要素类，参数 Allowed Geometries 就被激活了。

这是因为，Shape 严格限定了哪些几何类型能够通过特定的要素类（而 PDF 则没有限定）。

将参数 Allowed Geometry 设置为 shape\_polyline. 然后点击 OK。

**5) 重复转移过程**

对要素类“Rail”进行转移，然后运行工作空间，确保有两种不同格式的输出数据集。

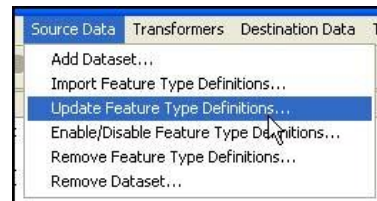


## 更新要素类

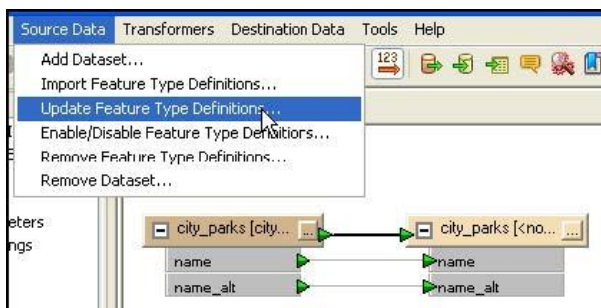
用户在处理空间数据时，会碰到一个常见的问题：在设置了数据处理任务后，源数据的结构却改变了；例如，出现了一个新的属性，或是已有属性由整型数变为了小数。**Workbench** 中的要素类定义会产生同样的问题，所以 **FME** 会提供一个工具来在新数据结构的基础上更新工作空间。

右图：选择菜单条中的 **Source Data > Update Feature Type Definitions**，就能够找到源数据的要素类更新工具。

**Destination Data** 菜单中也有一个类似的工具，更新目标要素类定义

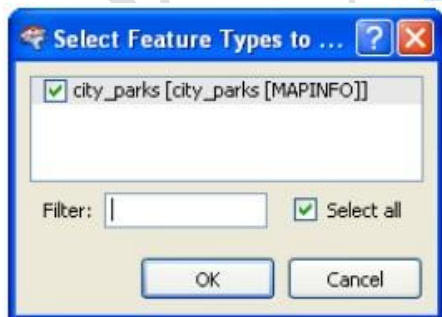
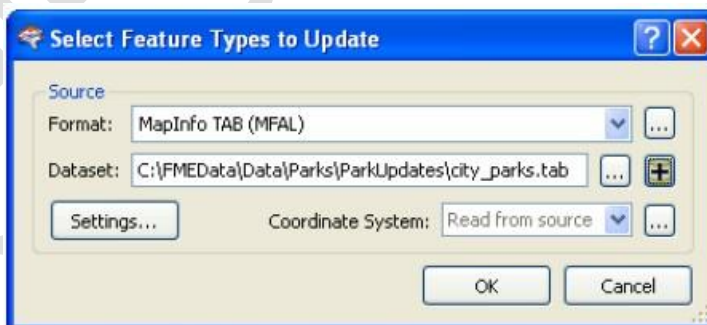


在下面的例子中，用户会将有关公园的数据从 **MapInfo TAB** 数据集转移到 **GML Simple Features** 数据集。在检查源数据的时候，他会注意到，因为之前已经创建了工作空间，所以源数据已经改变了，并且模式丢失了它的 **maintenance\_ID** 属性。



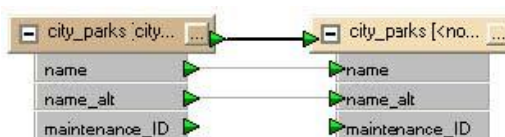
左图：工作空间显示了属性模式，并且用户即将选择更新要素类工具。

右图：用户选择更新的 **Mapinfo** 数据作为数据源进行结果更新。



左图：选择工作空间现有的要素类型进行更新。

右图：采取同样的方法，对 **GML** 目标数据集进行更新，并且使用相同的 **MapInfo** 作为源数据集，对它进行更新，现在工作空间就包含了正确的模式。

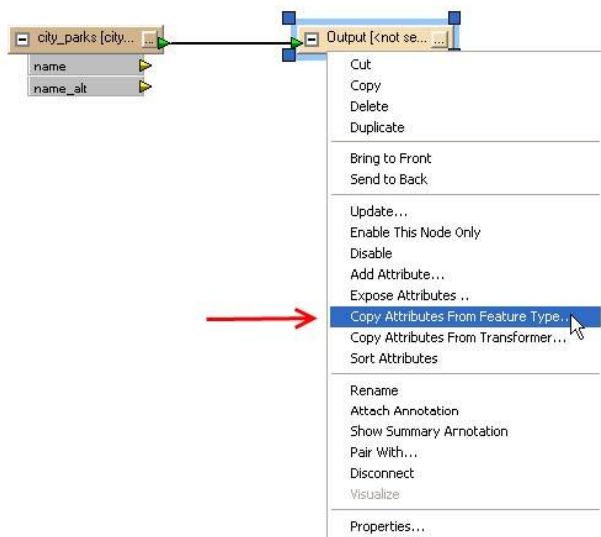
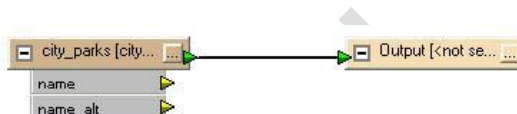


## 复制属性定义

有时候，有些目标属性缺少必须的属性，例如，由于从其它写模块中转移，或是从一个不同的数据集中输入，或是一个重新创建的属性，或者还没有随着工作空间更新新属性，都会导致目标属性缺乏必要的属性。

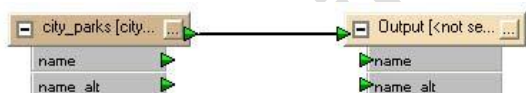
Worbench 就含有一个功能，能够从已有的工作空间中复制定义，这样就不必要手动地添加新的属性了。

右图：用户碰到了一个输出要素类，它的属性不正确。



左图：用户不需要自己输入属性定义，可以选择从已有的要素类中复制需要的属性定义。

右图：选择源要素类.....

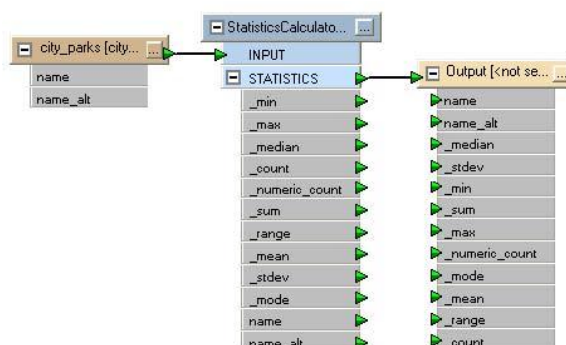


左图：Workbench 会完成剩下的工作。

右图：这个例子中，通过从 StatisticsCalculator 中复制属性，就能够很轻易地添加丢失掉的属性。



函数知道属性名，但是无法知晓属性类型，所有复制到目标要素类的属性都默认被定义为 char(20)类型，只有从其他要素类中复制属性，它才会有正确的属性类型。



**Example 5: 模式更新**

现在规划部门有要求对目标模式进行一些更改。

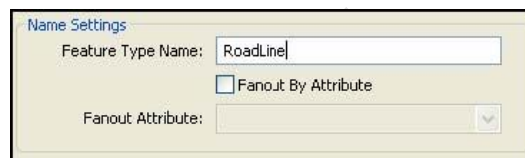
**1) 启动 Workbench.**

如果有需要就启动 Workbench，打开例 4 中的工作空间，或者重新创建一个，你也可以在 C:\FMEData\Workspaces\TrainingWorkspaces\S5 - Example 5 - Begin.fmw 中找到一个副本。

**2) 对 Output Feature Type 重新命名**

首先要做的就是，将 Road 要素类重新命名为“RoadLine”， Rail 要素类重新命名为“railroad”。

重新打开 Roads 的目标 Feature Type properties 对话框，将 Feature Type 重新命名为 RoadLine.，如左图所示。



重新打开 Rail 的 Feature Type properties 对话框，将 Feature Type 重新命名为 RailRoad。

**3) 更新输出模式**

然后，将目标模式更新到规划部门要求的标准。我们不需要添加属性值，仅仅需要更新输出模式，这样待会就可以输入模式值了。

幸运的是，在公司数据文件夹中，已经将模式保存在了已有的数据集中。

使用 Destination Data > Update Feature Type 来更新 RoadLine 要素类，然后使用以下数据集中的模式：

Schema Format	MapInfo MIF/MID
Schema Dataset	C:\FMEData\Data\Roads\RoadLine.mif

当出现提示时，选择“RoadLine”作为要更新的要素类，观察，现在目标模式是怎样拥有一套新属性的。

使用数据集，对 railroad 数据进行更新

Schema Format	Autodesk MapGuide SDL
Schema Dataset	C:\FMEData\Data\Railroads\railroad.sdl

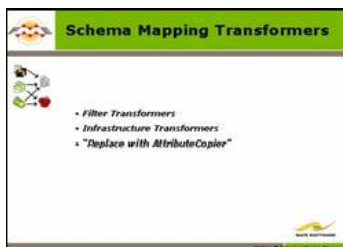
检查 railroad 要素类，确定已经对它进行了更新。

**4) 运行工作空间**

运行工作空间，检查输出—特别是 Shape 数据集。

怎样使用一个数据集中的模式来定义另一个数据集，这就是有关这个问题的一个很好的例子。

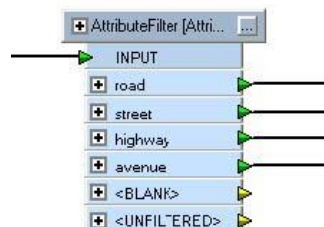
## 模式映射函数



Workbench 提供大量函数来帮助映射源模式到目标模式

### Filter 函数

这个函数检查要素几何属性，或者属性，并且根据检查结果，决定是否发送要素到不同的目标数据集



#### AttributeFilter

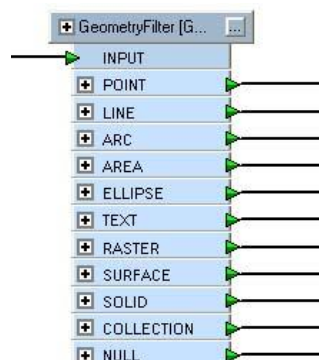
该函数根据选定属性中的特定值来分流要素。

左图：这个例子中，根据已选的属性值，将属性划分为‘road’，‘street’，‘highway’ 和 ‘avenue’。

### GeometryFilter

在几何类型的基础上，它能分流如下要素，例如，点，线，面，椭圆。

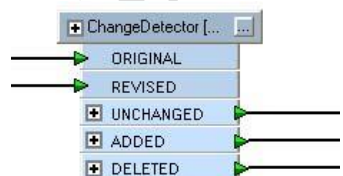
右图：这个例子中，输出数据被划分为多个几何流。



### ChangeDetector

通过比较，它能够检查出两组输入要素和并且根据比较的结果来分流数据。

左图：这个函数中，‘UNCHANGED’要素是存在于之前的几何流，以及修正的几何流中的要素：



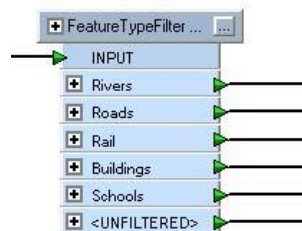
‘ADDED’要素就是只存在于修正后的几何流的要素，而‘DELETED’要素是只存在于之前的几何流的要素。

### FeatureTypeFilter

该函数在原来要素类的基础上，进行要素的分流。通过一个函数，也能够重新分离之前合并过的数据。

例如，不需要为每个要素类使用 AreaCalculator，你就可以将它们全部发送到一个单一的 AreaCalculator 中，然后使用 FeatureTypeFilter 函数，将它们再次分离成不同的数据流。

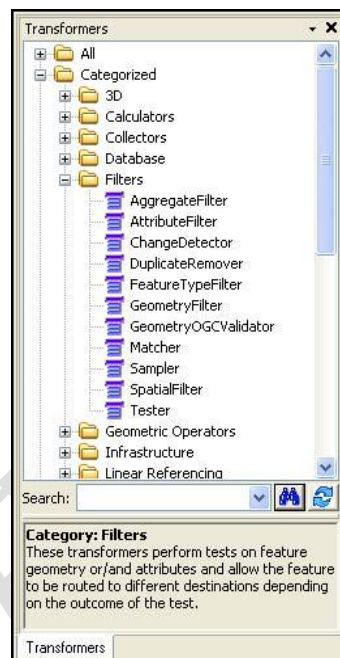
右图：将 Rivers, Rail and Roads, Buildings 和 Schools 要素合并起来，通过函数，然后再将它们分离成之前的要素类。





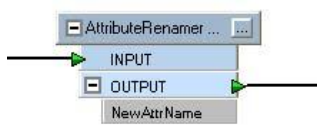
在 Workbench 导航方框中的 Transformer Gallery 中，会有一栏专门介绍 Filter 函数的分类。

右图：ransformer Gallery 中的 Filter 分类。



## 基本函数

使用一些基本函数就能够处理，调整属性，因此这些函数能更好地域目标模式相匹配，例如，复制或重新命名一个属性

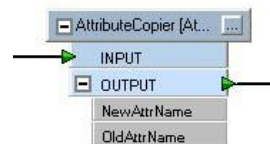


### AttributeRenamer

这个函数能够重新对一个属性进行命名。通过重新命名一个源属性，就能使它符合目标属性，即使是已经添加，删除或改变了要素连接，也能够自动地在工作空间中改变属性名。

### AttributeCopier

这个函数与 AttributeRenamer 函数相似，在新的属性名之下复制一个属性的副本，同时仍保留之前的属性。



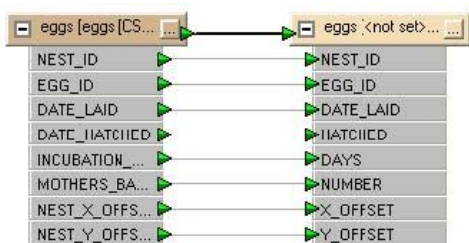
例如，AttributeRenamer 与 AttributeCopier 之间的区别是，在 AttributeRenamer 函数中会清除之前的属性，但是在 AttributeCopier 中仍然保留了之前的属性。



## Replace with AttributeCopier

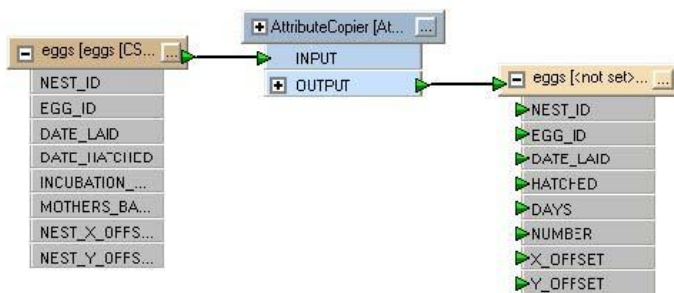
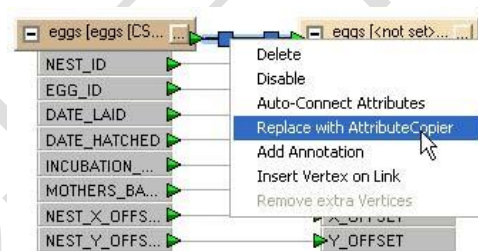
你的讲师可能已经指出了，使用 **AttributeRenamers** 函数能够更好地进行属性映射，因为它们持续的，并且无论工作空间怎么改变，仍然能够保留之前的设置，虽然如此，用户仍然更喜欢进行手动映射，因为这样会显得更加直观。

**Replace with AttributeCopier** 功能就能够让用户将这两种方法结合起来，使用 **AttributeCopier** (包含同等的属性映射)，就能够取代之之前手动创建的属性映射。



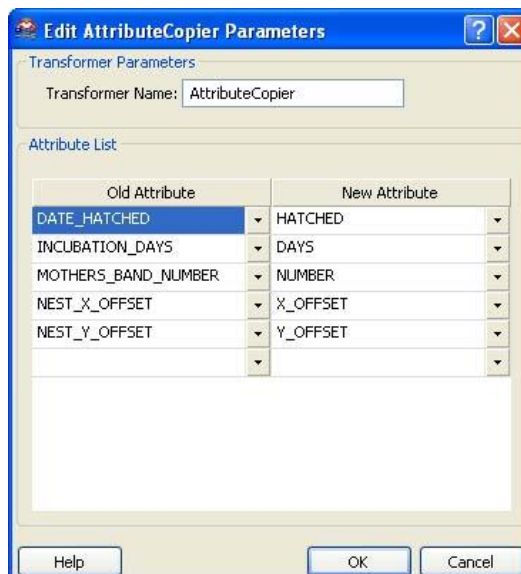
左图：对目标数据集的一组属性映射。

右图：右击主要的要素连接，然后选择‘Replace with AttributeCopier’



左图：手动连接就被 **AttributeCopier** 取代（它执行相同的映射）。观察，为什么目标属性的输入端口显示成绿色，这就表示有一个隐藏的连接。

右图：AttributeCopier 设置已经自动生成了。



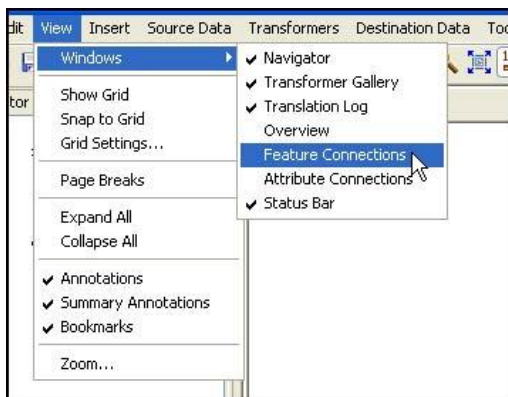
## Workbench 模式映射功能



除了使用函数，Workbench 有其它的功能，帮助将源模式映射到目标模式。

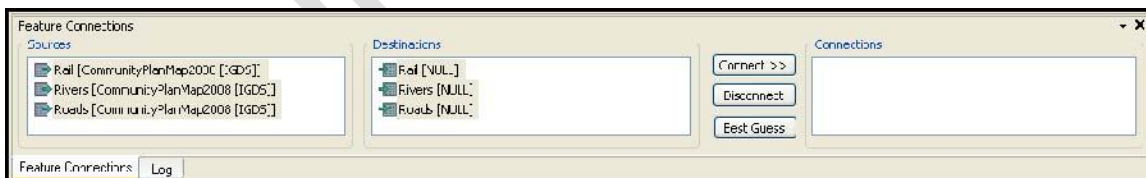
### 要素连接方框

使用它能够快速将要素类和函数连接起来。当你需要进行大量连接时，这个方框能为你节省大量的时间。

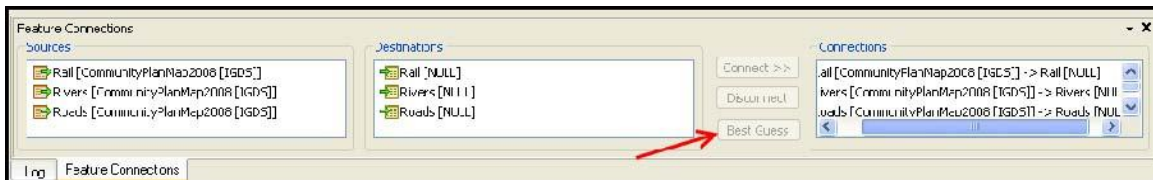


左图：一般要素连接窗口是关闭的，所有就要从菜单条中选择 View > Windows > Feature Connections，打开这个窗口。

下图：在这个简单的例子中，已经选择了三个源要素类和三个目标要素类，点击链接键，就可以将源和目标要素类进行连接了...



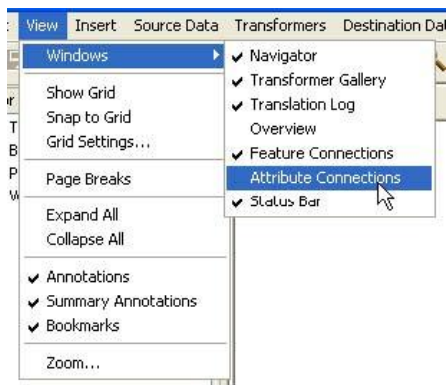
...但是更合适的操作是，使用 Best Guess 键，这样 FME 就能通过要素类名决定要进行哪些连接。





## 属性连接方框

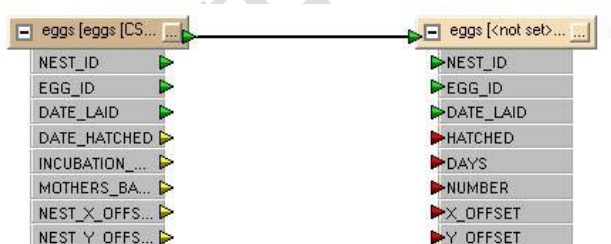
使用它就能够快速地将要素类的属性和函数进行连接。当要进行大量的连接时，使用这个方框能为你节省大量的时间



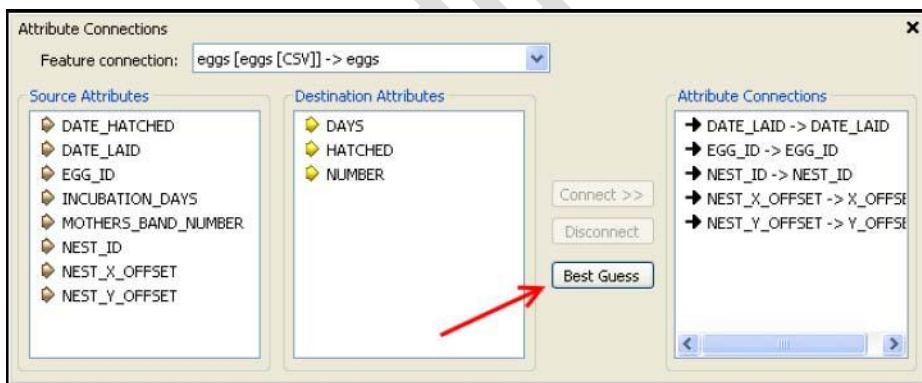
左图：一般属性连接窗口是关闭的，所有就要从菜单条中选择 View > Windows > Attribute Connections，打开这个窗口。

右图：这个例子中，使用简单的方法来进行属性连接。

已经调整了目标属性名，来满足格式的要求，并且大多数属性需要进行重新连接。



可以使用属性连接窗口来进行连接，这样就能同时进行多个连接了。



上图：Attribute Connections 窗口中的 Best Guess 选项已经配对大量拥有相似名字的属性，这样用户就能够手动地处理其他的属性了。

## 单元复习



这个单元帮助你进行多个数据集转换，添加新的数据集和要素类到已有的工作空间。

你应该从这单元中学到些什么？

下面就是你应该从这单元中学到的主要内容：

### 理论

- FME 能够以任何要求组合的方式，进行多种数据集转换，多种格式转换。
- Unexpected Input Remover 是一个工具，使用它就能够过滤没定义的要素类，以及对源要素的读取进行控制
- 可以将模式理解成一个独立的实体，具有独立的格式，并且能够应用于任何读模块和写模块。

### FME 技能

- 能够创建一个工作空间，转换多个数据集和要素类
- 能够在已有的工作空间中添加一个新的数据集和要素类
- 能够输入要素类到工作空间，并且使用一个 merge feature type filter.工具，能够选择要通过的要素类
- 能够处理一个模式，并将它运用到不同的数据集
- 使用 Filter 和基本函数就能够映射一个模式

## 疑难解答



### Q+A Answers



Miss Vector 说过...

“下面就是测试的答案。我们学到的内容很重要，所有一定要与培训师讨论你不明白的问题。”

Import Feature Type Definitions 和 Merge Filters 解决了哪些问题？

- 1) 读取不同属性的数据集
- 2) 读取不同要素类的数据集
- 3) 读取不同格式的数据集
- 4) 读取不同几何类型的数据集

<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

一般来说，这两个都能解决不同要素类的问题，但是 Import Feature Type Definitions 还能够解决那个属性的问题。

Import Feature Type Definitions 执行哪些操作？

- 1) 添加丢失的要素类
- 2) 允许要素类通过
- 3) 关闭 Unexpected Input Remover
- 4) 更新已有要素类的属性

<input checked="" type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

Merge Filter does 进行哪些操作？

- 1) 添加丢失的要素类
- 2) 允许要素类通过
- 3) 关闭 Unexpected Input Remover
- 4) 更新已有要素类的属性

<input type="checkbox"/>
<input checked="" type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

你是怎么看待它们的，例如，你认为每个的优点和缺点分别是什么？

Import Feature Type Definitions 能够分别保存工作空间中的每个要素类，对要素类和它们的处理过程分别进行控制，缺点就是：在进行转换之前，一定要定义所有的要素类。

虽然这个并不是强制而明确的规则，但是，总的来说，使用 Import Feature Types Definition 来处理文件形式的数据集，例如，MicroStation DGN 或 AutoCAD DXF 格式。

Merge Feature Type 工具则不需要提前知道所有的要素类。不管是否知道要素类，只要设置针对所有要素类的过滤器为 “\*” 的通配符，就能处理源数据的所有要素类。但是，在发送所有数据到单一的要素类时，就不能对处理每个要素类进行分别控制了。

当然这并不是绝对的，我们通常在文件夹形式的数据集中使用 Merge Feature Type，例如，Shape，这样就能确保处理数据集中的所有要素类。

为什么你会认为 FME 拥有一个 Destination Data > Add Feature Type Definition, 而不是一个 Source Data > Add Feature Type Definition?

源数据的区别是什么，它阻止数据使用这种工具。

答案：目标模式是“我们需要什么”，所以就合理地编辑这个模式，并且按照需要添加要素类。而源模式是“我们有什么”，所以，大多数情况下，我们不能任意地向源数据集添加要素类，因为要素类在源数据集中并不存在。