目录

简介 3
获取连接....3
数据库 Reader 的最优方法 6
使用地理数据库....8
Reader 参数 8
Writer Parameters 参数....11
Feature Type Level Parameters 参数 13
注记提示和技巧（Annotation Tips and Tricks） 18
读入注记（Reading Annotation）....18
写入注记（Writing Annotation）....19
要素相连的注记（Feature Linked Annotation ）....21
操作关系类....24
读取关系类....24
写入关系类....25
读取和写入几何网络....32
什么是几何网络？....32
写入到几何网络....32
从几何网络中读取....32
元数据（Metadata ）....35
什么是元数据？....35
更新数据....38
Writer Mode....38
要素级别的更新（Feature Level Updates）....41
对标签要素使用格式属性....41
变化检测....42

北京世宏安图

变化检测.....42
Transformer 更新.....46
SQLExecutor.....46
ArcSDEQuerier.....46
数据分发.....47
进行数据分发的好处.....47
简单的输出数据.....48
针对其他用户的工作空间.....50
转换的可能性（Transformation Possibilities）.....52
使用 FME Server 进行数据分发.....53
工作空间的考虑 （Workspace Considerations）.....53
发布一个工作空间.....53
学期回顾（Session Review）.....54
从这个单元中你能学到什么？.....55
附录 A – 兼容性和安装问题.....55
附录 B – 有用的调试资源.....56
附录 C – 性能考虑.....56

北京世纪安图

FME 对 ESRI 发烧友的支持

简介

组织或公司为了保证对空间数据的访问和数据库的质量的控制，他们通常将数据集中到一个空间数据库中。这通常包括把数据从中心数据库的迁入与迁出，以便完成每天的工作任务。这里我们将介绍使用 ESRI 的地理数据库格式（GeoDatabase）来储存数据，以及如何在这一数据上进行数据的加载和提取。

在本课程中，我们将演示用 GeoDatabase 加载更复杂的要素类的方法。我们将以一个文件/个人 SDE GeoDatabase 开始，这个数据库中包含一些简单要素层，然后在这些数据上做些例子，包括加载其它的简单数据、与要素关联的注记、关系类以及几何网络。

接下来我们将介绍从 GeoDatabase 中提取数据的一些方法——产生 CAD 图或通过网页界面来发送它们。

学习本课程的前提条件

学习者需对 FME Desktop、 创造工作空间以及更多的常见的 FME transformer 有一个了解。这里我们不会讲解 GeoDatabase 的创建和维护——这些信息可以从 ESRI 公司获取。

软件要求

FME 提供了读写器，用于访问存储在 ESRI 模板库中的数据。GeoDatabase 的 reader/writer 需要调用 ESRI ArcObjects 类库，因此需要授权的 ArcGIS 来支持 GeoDatabase 的格式：

- 个人 GeoDatabase（文件和 MDB 格式）
  - O 读出过程：ArcView 或 ArcEngine 级别或更高版本
  - O 写入过程：ArcView 或 ArcEngine 级别或更高版本
- 个人或企业 SDE GeoDatabase
  - O 读出过程：ArcView 或 ArcEngine 级别或更高版本
  - O 写入过程：ArcEditor 或带 GeoDatabase 扩展更新的 ArcEngine

FME 也包括 ArcSDE Reader/Writer，这不需要安装 ArcGIS，不过访问 SDE 类库就需要安装 ArcGIS。这些可以从 SDE SDK 的安装、ArcGIS Desktop 乃至 ArcGIS Explorer 的安装中获得。这个 SDE 的 reader 和 writer 也需要有能访问的 ARCGISHOME 来指向一个文件夹，这个文件夹包括必需的.dlls 文件的\bin 文件夹。

更详细的信息看附件 A。

另外，针对 GeoDatabase/SDE 等的操作，只有 FME for ESRI 版本或更高版本中才有。

获取连接

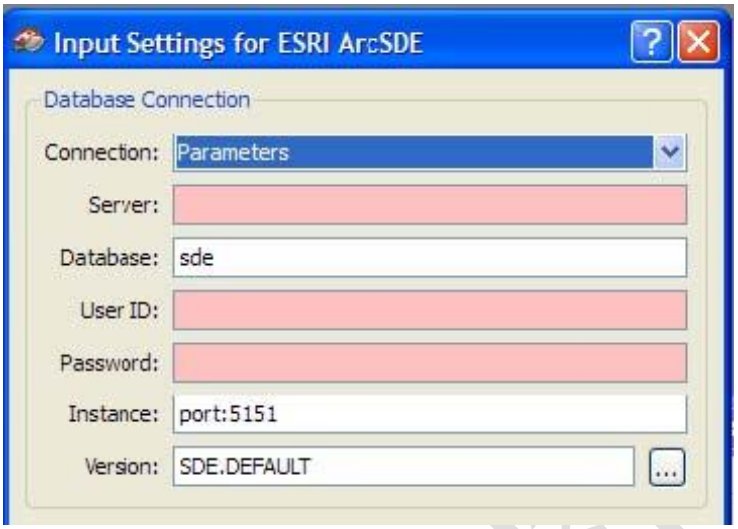
使用 FME 进行 GeoDatabase 的数据转换不止是要 FME 知道文件的位置。GeoDatabase 可能包含了许多要素集、要素类、表等等，它可能存在于企业环境服务器中，或是使用 Access 作为文件存储方式的个人地理数据库中。

对于我们的这个课程，我们将使用个人的基于文件的 GeoDatabase（PFG）和 ArcGIS Desktop 9.3.1。我们已安装了一个个人 SDE GeoDatabase，它用来展现只有在 SDE GeoDatabase 中可获取的选项。

因为许多人实际上是在网络环境中使用 GeoDatabase，因此值得我们花几分钟来看一下访问 SDE 空间数据需要的参数。每次要访问 SDE GeoDatabase，都需要你指定一种方法来连接这个数据库。我们通过设置 Reader 或 Writer 上的设置框来进行连接设置。

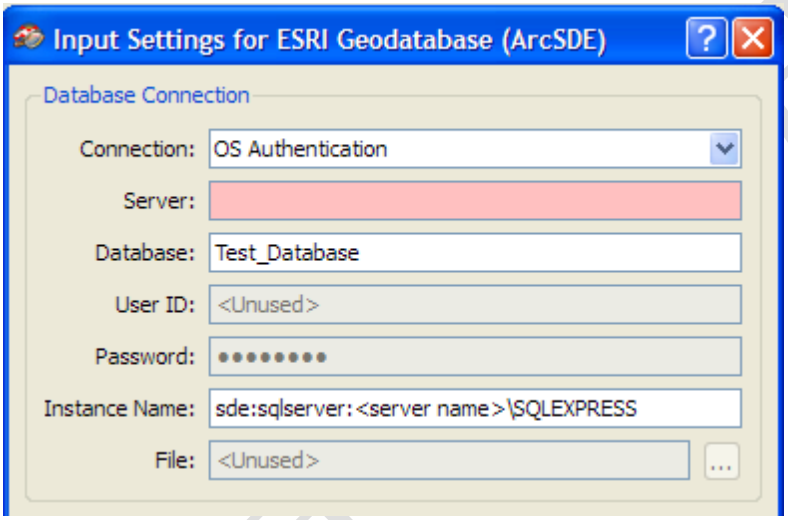
连接参数

SDE 的连接设置（右图）会随着你使用的 Reader/Writer 的类型和连接方式而变化。



操作系统认证 （直接连接）

如果使用 2 层(直接连接)方式访问 GeoDatabase，你必须选择 OS Authentication 连接设置。这种情况下，Windows 会为连接提供系统级认证。这种方法也用来连接到个人 SDE Geodatabase。

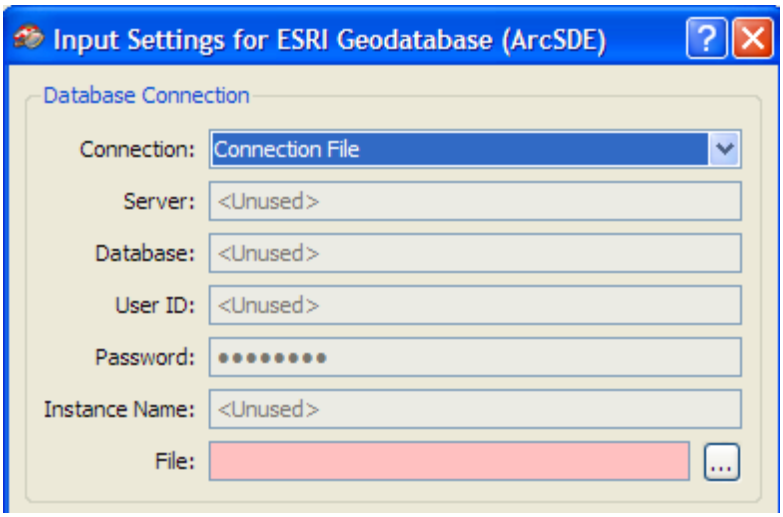




注意：即使我们要求你填写一个服务器，实际上我们并不使用它。这来自对于常规的操作系统认证数据库时使用的相同的设置框。

连接文件

指定连接设置的另一种的方式是使用连接文件。这个连接文件必需要是一个*.sde 文件，并且是 ESRI 定义的连接文件的正确格式。

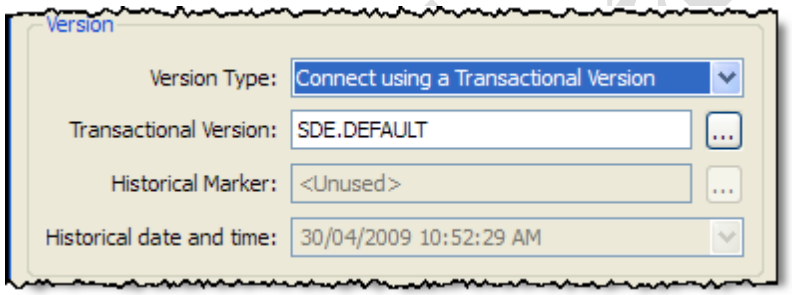


连接陷阱！（Connection Gotchas）

可能有许多原因导致你不能连接到数据库。查看 FME Readers and Writers 手册，在 ESRI ArcSDE Reader/Writer Troubleshooting 部分有一些普通问题的信息。

版本详情

在设置对话框里的下面这组设置，决定你是否能从版本数据库中读取数据，以及你想从哪个版本中读取。

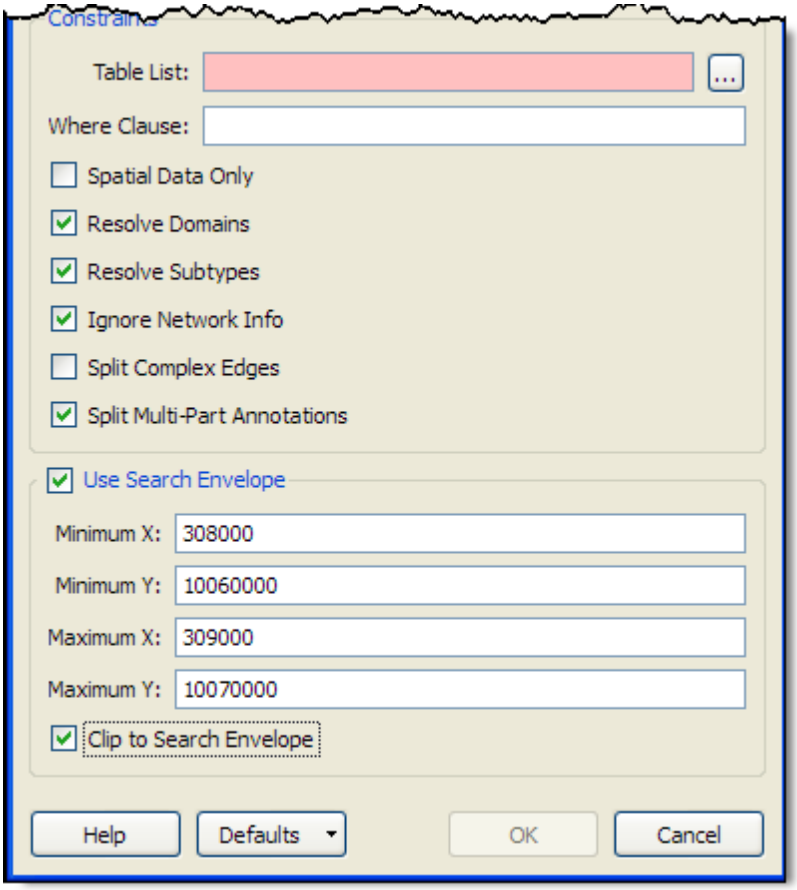


上图：GeoDatabase（ArcSDE）reader 版本信息设置

个人 SDE 和其它的 Windows 认证数据库通常都有一个 dbo.DEFAULT 的默认版本。通过点击 Transactional Version 选项后面的“…”按钮就可以选择正确的版本。

限制（Constraints）

在 Constraints 部分中，你可以指定你想要读入的要素类，任何可能应用到要素类中的 WHERE Clause 和 search envelope 设置等等。



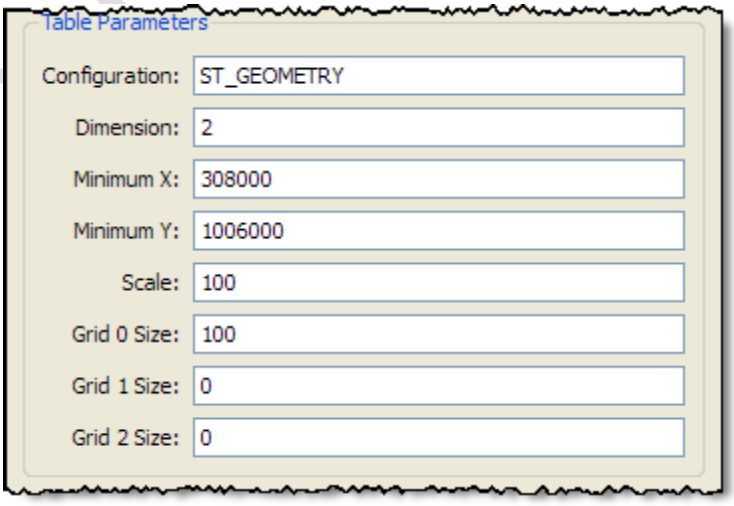
左图：GeoDatabse（ArcSDE）的 reader 设置，用来决定各种要素类如何被读入以及定义 search envelope。

数据表参数（Table Parameters）

只有在 GeoDatabse 或 SDE 的 writers 中才有空间数据表参数设置对话框。在这些参数中可以定义数据的空间范围，也可以指定对于你想写入的要素类的不同的配置设置。

上图：ArcSDE 数据库 writer 的附加参数。这些参数包括数据表维数（2D 或 3D）和定义表空间范围的项目。

这个配置设置是指向 dbtune 表中的 ST_GEOMETRY 实体的例子。



在 writer 上填写所有的空间范围域确实是个很好的习惯。如果你不填写的话，数据库仍会被创建，但是不能再更改这些限制条件。

这个数据可能仍然会加载，但是其他的数据库功能，例如索引，可能不会如你期望的那样执行。花点时间来正确设置它吧！

数据库 Reader 的最佳实践

最佳实践意味着我们会尽可能多的把数据处理留给数据库，因为对于这些任务，数据库的工具是最优化的。而且我们也不希望读入太多不必要的数据。

记住这一点：

何时使用条件过滤（Where Clause）

当你打算通过一个指定属性值来过滤数据时，就使用 Where Clause。

比起读入整个数据集，然后使用 FME Tester 来移除不需要的数据，一个 Where 条件更高效。



对于 GeoDatabase 或 SDE 读模块指定一个 Where Clause，会导致 WHERE 从句对这个特定的读模块中的每个要素类。这可能导致读模块的性能问题。有时候添加一个不带 Where 从句的额外的读模块来读取这些不需要 where 从句的要素类是必要的。

何时使用范围搜索（Search Envelope）

当你想要在空间上过滤数据时，使用一个 Search Envelope（搜索范围）。

比起读取这个数据集，再使用 FME Clipper 来在空间上过滤数据流，一个 Search Envelope 更高效。



例 1——在 FME Viewer 中浏览 GeoDatabse 或 SDE 要素类

让我们花点时间来浏览已经加载到 GeoDatabse 中的数据

1） 打开 FME Universal Viewer

如有必要，就启动 FME Universal Viewer，选择“open dataset”工具。

2） 设置连接参数

连接参数的设置如下：

Source Format　ESRI Geodatabase (File-based)

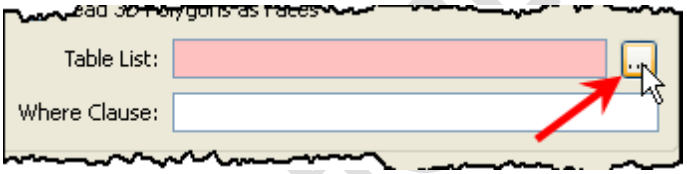
Source Dataset　C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

Source Settings:

Table List:　select all the feature classes

3） 读取所有数据

首先，读取数据库中的所有数据。



点击 Table List 选项右边的[…]按钮。就会出现读取到的数据表目录。然后选择所有的表。

点击 OK 按钮，观察数据。



小提示：我们不再有“Deselect All”按钮，因为“Select All”工具现在是一个“三态”（“tri-state”）的按钮！

4） 只读取一个 City Park 要素

现在使用条件过滤（where clause）参数，只读一个 City Park 要素。

再次打开数据集设置对话框。点击 Table List 选项右边的[…]按钮，就会出现读取到的数据表目录。选择 CityParks 表。在 Where Clause 选项中输入查询语句：

NAME='Bartholomew'

注意，字段名是不区分大小写的，要用单引号(')，而不是双引号(")

点击 OK 按钮，就会看到 FME 已经从数据库中读取了一个公园要素。

5） 使用 Search Envelope

现在我们来尝试使用 Search Envelope。

再次打开数据集设置对话框，在 Table List 中选择所有表。

删除 where clause，并把“Use Search Envelope”这一项中勾选。

参数设置如下：

Minimum X: 3127500 Maximum X: 3130750

Minimum Y: 10083000 Maximum Y: 10085000

重复上述过程，这次选中“Clip to Search Envelope”选项。这样做会产生什么不同？

使用 GeoDatabase

Reader 参数

有许多 Reader 级别的参数（Reader level parameters）。这些参数影响数据的读取方式，对 reader 的性能也有些影响，所以了解这些参数是有用的。



左图: Geodatabase（文件）reader 参数。

条件过滤（Where Clause）参数：

Where Clause 用来选择从 Geodatabase 中提取特定记录。这个条件过滤（Where Clause）几乎可以像一个 SQL 从句（使用基本数据库支持的语法），除了 ORDER BY 语句、GROUP BY 语句、嵌套查询和聚合函数（例如 MAX, COUNT)不能被使用外。

Spatial Data Only 参数

当设置为 YES，非空间表、关系、域名和子类型不会被读取。当创造一个工作空间或映射文件时，如果这个参数被指定，对于非空间表不会返回结构模式。

Resolve Domains 参数

这个参数指定是否处理含有与属性相关的编码值域的属性（默认域，或是被子类型创建的域）。这意味着当要素的属性有一个与它相关的编码值域时，我们需要添加另一个属性来表达编码属性的文本描述。这个新的属性是<attribute-name>_resolved，其中<attribute-name>是包含编码的属性名。只有当<attribute-name>包含一个非空（non-NUL）值时，我们才需要添加这个属性。默认的不选择这个参数。



如果你的要素类已经有了大量的属性，但你需要确保域信息，就使用 AttributeRemover 从工作空间中移除不必要的属性。

Resolve Subtypes 参数

这个参数用来指定是否分析一个要素的子类型域。在表中有子类型的要素有一个子类型域的属性。这个子类型域含有一个整数值指定这个要素属于哪个子类，这个整数值也有一个相当于描述的字符串名。这个参数默认是选中的，所以相应的描述会作为一个属性加在要素上，这个属性是 geodb_subtype_name。

Ignore Network Info 参数

这个参数决定是否读取网络要素的网络信息。

如果设置成 NO，地理数据库指定的描述网络信息的属性，如网络连通性，就会插入到这个要素上。但不管这个参数值是多少，要素的几何都是相同的。默认值（NO）是读入网络几何，所以联接点将被作为点要素，边就作为多线段要素（polyline），同时将 geodb-type 分别设置成 geodb-point 和 geodb-polyline。

Ignore Relationship Info 参数

这个参数决定是否读取存在于源数据集中的关系要素。

当设置成 YES，含有简单关系（simple relationship）的要素类会被忽略，含有属性关系（attributed relationships）的要素类会作为非空间数据表处理。如果设置成 NO，不管是简单关系还是属性关系都会正常读取。



如果忽略网络（network）和关系（relationship）信息——也就是，将 Ignore Network Geometr 和 Ignore Relationship Info 都设置成 YES，读取网络要素和关系要素的速度会大大改进。

Split Complex Edges 参数

当设置成 YES，复合边（complex edges）会被分割，每个被分割的要素储存描述它两端的连接点的属性和组成复合边的其它各个要素的次序。

Search Envelope 参数

当你想使用矩形空间范围（Minimum X, Minimum Y, Maximum X, Maximum Y）来过滤数据时，设置 search envelope 选项。这个参数可以与查找方法（Search Method）结合。

Clip to Search Envelope 参数

当设置成 YES，与 search envelope 重叠的要素就会剪切到这个搜索范围。

Split Multi-Part Annotations 参数

这个参数决定是否在读入时将组合注记（multi-part annotations）分割成对每个‘元素’的单独要素。如果设置成 yes，对于在组合注记（multi-part annotation）中的每个元素（通常是一个词）的单个要素会产生一个读取过程，导致指定的要素属性如角度和文本位置会根据每个成分的位置被保存。设置成 NO，组合注记（multi-part annotations）就会被正常读取，作为储存一系列描述文本分布属性的单个要素。

Search Envelope Coordinate System 参数

如果查找范围的坐标系统与数据的坐标系统不一致，这个参数就指定查找范围的坐标系。这个坐标系统参数指定了读取与数据相关的坐标系统，如果这个参数被设置，就要始终设置它。

Search Feature 参数

这个参数提供了一个机制来指定一个任意的查找要素（如输入坐标 <xcoord> <ycoord>…的列表，定义一个几何查询）。这个参数与查找方法（Search Method）参数一起使用。

Search Method 参数

这个参数指定了用空间过滤查找时的空间关系的类型，被查询空间要素必须要有查找范围（Search Envelope）或是查找要素（Search Feature）才能得到正确的结果。有效值可以是 GEODB_INTERSECTS, GEODB_ENVELOPE_INTERSECTS, GEODB_TOUCHES, GEODB_OVERLAPS, GEODB_CROSSES, GEODB_WITHIN, GEODB_CONTAINS。

Search Order 参数

指定了在 Geodatabase 上执行的基本搜索次序。这个参数决定是首先查询空间部分（spatial component）还是属性部分（attribute component）。使用这个参数的好处是提高效率。例如，如果属性部分能比空间部分更好的过滤数据，那么可以使用 Search Order 参数强迫首先使用属性部分。可能值是 SPATIAL_FIRST 和 ATTRIBUTE_FIRST。

Feature Read Mode 参数

当设置为元数据后，这个参数提供了读入表级别（table-level）元数据的能力。当设置为要素时，这个 reader 输出存储在表中的要素。

Check for Simple Geometry 参数

这个参数指定是否要对地理数据库中的要素进行检查，从而判定它们是否是简单的。这个过程会非常费时，并且会影响 reader 的性能。

Merge Feature Linked Annotation 参数

这个参数指定，当读入时，在注记连接的主要素，是否把关联的注记（feature-linked annotations）的文本、角度和位置等属性合并到主要素上。如果该参数设置成 YES，则对于注记会产生一个属性组，包含在所有注记的属性中，而不去读取注记表。设置成 NO, 就会将连接要素的注记（feature-linked annotation）作为常规注记读取。

Make Connection Persistent 参数

它用来指定是否创建一个与 SDE 的连接，这个连接持续存在，可以被其它的 Geodatabase 的 Readers 和 Writers 共享。设置成 YES，只有关闭 FME，连接才会断开，即使是 reader 已经停止了使用它。设置成 NO，关闭了 reader 连接就会断开，除非其它的 reader/writer 仍在进行连接。

Archive Where Clause 参数

这个选择参数指定 where 从句限制从归档表中读取的要素。日期数据一定要是 FME 的日期数据格式，并且能转换为基础数据库要求的格式。

Writer 参数

Writer 级别的参数会影响整个数据集。

因此，就 Geodatabase 来说，Writer 参数会影响到写入 Geodatabase 的所有数据。



Overwrite Geodatabase 参数

这个参数会导致 writer 删掉原来的 Geodatabase，然后使用工作空间正写入的要素类来重建它。只有要取代整个 Geodatabase 时，才会使用这个参数。正是因为这个原因，默认参数值是“NO”。

Writer Mode 参数

另外一个我们需要知道的关键的 Writer-level 参数就是 writer 模式（Writer Mode）。

这个参数值可以从以下三个中任取一个：INSERT, UPDATE 或 DELETE。

INSERT 是默认值，它用来将数据插入到地理数据库中。正如你所猜想的，INSERT 格式的 writer 只会简单地将所有要素添加到数据表。如果表中要素已存在（例如，复制 ID 号到已有的要素中），不会进行更新，只是会插入记录。因此，复制一个存在的记录或者如果 ID 字段被定义为一个唯一索引，就会产生一个警告并删除这条记录。

然而如果你正在一个已存在的数据库上进行操作，或许你可能对更新或是删除无用信息更有兴趣。使用 UPDATE 模式的 writer，我们不仅能根据需求更新要素类中的信息，如果需要，也能插入、更新信息。

Transaction Parameters 参数

在 Geodatabase 中和 SDE 上的 Writer 都是事务处理参数（Transaction Parameters）。

Transaction 就是分批地将数据编写到数据库中，这样可以提高性能，也可以避免因单个

坏的要素而导致的应用程序的异常终止。

Transaction 参数指定是否使用事务处理功能以及每组写入多少个要素。

Transaction Type 参数

这个选项指出了 Geodatabase 的 writer 应该使用的是哪个事务处理机制。在 ArcGIS 中，现在有两种事务处理机制：编辑会话（edit sessions）和（常规）事务处理（regular transaction）。

Features to Write Per Transaction 参数

这个参数告诉 FME，在提交处理事务到数据库前，放置在每个事务处理（transaction）中的要素的数目。

Transaction Number 参数

假设因为一个错误的要素导致处理事务失败，比如第五个处理事务，用户就可以在数据上修复这个错误，重新运行转换，指定从第五个事务处理开始写入。我们知道最开始的四个 transaction 是正确的，就不需要再次写入数据库。这样的话，在转换时就不需要再花时间来重新写入数据。

Transactional Version 参数

它指要素应该写入的版本的名称。

Reconcile and Post 参数

这个可选参数协调被 Writer 参数 VERSION 指定的版本和它的父版本之间的所有变化，甚至是在当前转换外的编辑操作。然后发送这个版本到父版本。在转换的最后，当关闭地理数据库的 writer 时，调整和发送才被执行。结果，如果在协调（reconcile）或发送（Post）过程中发生错误，就只有协调和发布的更改还没有做完；之前的所有插入、更新、删除的要素都已经被保存了。如果没有发现冲突，协调和发送（reconcile and post）才会成功。冲突必须使用 ArcGIS 手动解决。在成功发送过程中，根据 Delete Child State Following Reconcile and Post 参数的值，来删除子版本或是维护原样。

Delete Child State After Reconcile and Post 参数

这个参数决定是否删除追随协调和发送的子版本，即使母版本和子版本一样。

Make Connection Persistent 参数

它指定是否创建一个与 SDE 的连接，这个连接能被保持并且可以被 SDE Readers 和 Writers 共享。设置成 YES，即使这个 reader 已经不再使用它，连接仍然是打开的，直到关闭 FME。设置为 NO，关闭 reader，连接也就断开了（除非是其他的 reader/writer 仍然在使用这个连接）。

Simplify Geometry 参数

如果设置成 YES，几何被简化后写出（如果当前是非简单几何）。

Ignore Failed Features 参数

这个参数会告诉 Geodatabase 的 writer 是否要忽略掉一些通常会引起转换失败的要素。它允许你忽略拓扑错误的要素、不被 Geodatabase 的 writer 支持的要素或是与将要插入到的数据表的定义冲突的要素（即，这些要素不在要素类指定的几何范围内）。另外，会忽略掉不能重定向的多边形（polygons）、圆环（donuts）、或多边形/圆环的聚类（aggregates of polygons/donuts）。

Max Number of Features to Ignore 参数

这个参数告诉 Geodatabase 的 writer，在由于一个有问题的要素引起转换失败之前，要忽略的要素的号码。虽然如此，由于其它的原因转换仍然可能失败。这个参数只适用于将 Ignore Failed Feature 参数设置成 YES 的情况。

Dump Failed Features to File 参数

这个参数允许用户保存失败的要素到 FFS 文件，以便以后浏览。必需设置 Ignore Failed

Feature 参数为 YES，。

Failed Features Dump Filename 参数

这个参数允许用户选择将含有错误要素的 FFS 文件放在指定路径下。这个文件可以自动的创建，但是只有存在失败的要素时才会自动创建文件。如果用这个参数保存的文件与其它文件同名，就会覆盖已有的文件。必须设置 Dump Failed Feature 为 YES，才能应用这个参数。

Annotation Units 参数

这个参数允许用户在创建新的注记要素类时，指定使用的地图单位。如果 Writer 创建了一个注记要素类，这个参数被设置为 unknown-units（默认值），那么 writer 试图确定使用的空间参考的单位，设置注释单位为一个稍大的或等于它的单位的最接近的单位（比如说每个单位值为米）。如果使用的是一个本地的或未知的坐标系统，单位会被设置为米。

Compress Database When Done 参数

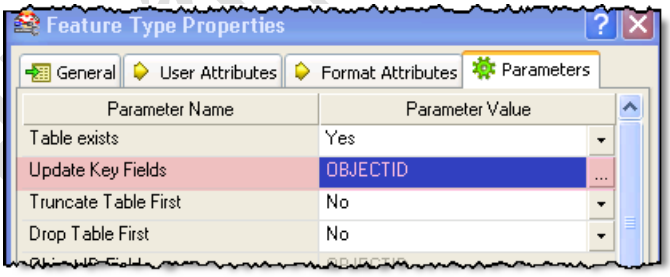
这个参数决定在 Writer 完成了对要素的写入后，这是否要压缩数据集。它对个人的（personal）和文件类型的数据库都适用。

Feature Type Level Parameters 参数

要素类型层次参数(Feature Type Level Parameters)是影响单个的要素类型(Feature Types)的一类参数。在 Geodatabase 中，每个 Feature Type 代表单独的数据表，因此参数仅仅影响写入到指定数据表中的要素。

Update Key Fields 参数

当 Writer mode 是 UPDATE 时，指定哪些将要写入的要素将更新哪些存在的记录是必须的。



这个更新关键字域（Update Key Fields ）（左图）是用户定义连接要素和记录的属性的方式。

Drop/Truncate Table 参数

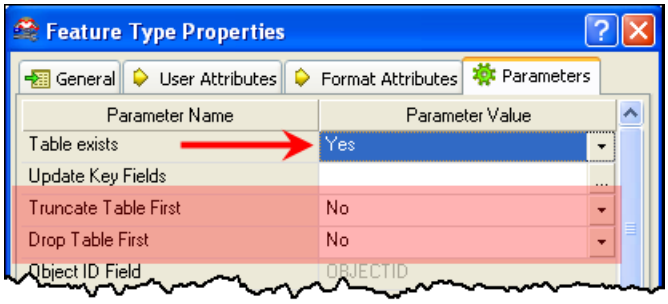
当向一个已有的 Geodatabase 表写入数据时，可能想清空这个表或彻底删除它。Geodatabase 和 ArcSDE 的 Writer 都有这个参数，让你在要素类层次（Feature Type Level）上执行这项操作。

当使用这两个选项时，它们的不同如下：

Truncate: 清空数据表是把它的内容清空。在你需要替换表中所有数据时，这项操作会非常有用，但是数据表结构不会改变。

Drop: 删除一个数据表是从数据库中删除它。在你需要替换表中所有数据时，这项操作会非常有效，但是也替换了数据表的定义。

右图: 这个 Geodatabase（SDE）的要素类属性对话框中有一个选项决定在写入数据前是清空还是删掉表。



注意只有将 Table Exists 参数设置成 YES 时，才会显示这个参数。如果表不存在，那么 Truncate 和 Drop 参数就会呈现灰色。

请参阅 FME Readers and Writers 手册，你可以了解更多关于这些参数如何在个人的和企业级 Geodatabase 以及 Geodatabase 的 Writer 和 ArcSDE 的 Writer 上设置。

例 2 —加载数据

通过一个简单的例子，我们来看一下将 Access 数据库中的数据加载到 Geodatabase 中。有一些 AddressPoints 数据，在后面的例子中会用到。

1) 创建工作空间

创建一个工作空间来加载 Address 的点信息到要素数据集中。源数据就存储在 Access 数据库中。工作空间设置如下：

Source Format： Access .MDB file

Source Dataset ： C:\FMEData\Data\Addreses\roadAllowancesAndAddressPoints.mdb

Source Settings:

Table List: ADDRESS_POINTS

Destination Format ： ESRI Geodatabase (File-based)

Destination Dataset： C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

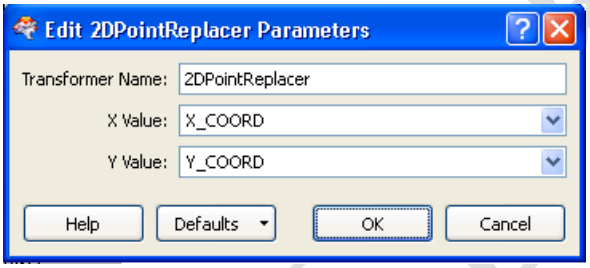
Destination Settings

Table exists: No

当使用数据库格式时总是检查 reader 和 writer 的设置框。这通常用来一次性设置一些可选项。

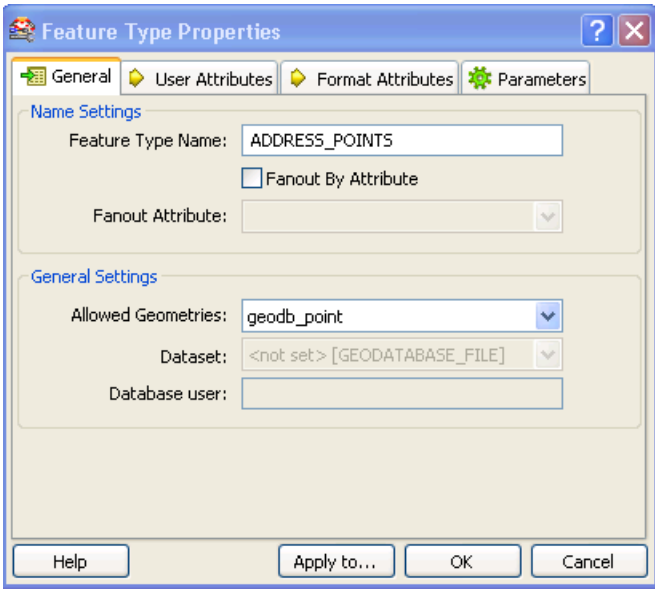
2) 放置一个 Transformer – 2DPointReplacer

我们需要转换这一非空间数据为点数据，所以添加一个 2DPointReplacer 转换器，使用 X_COORD 和 Y_COORD 属性来创建这个几何。



3) 检查目标数据设置

在创建工作空间时，目标数据集的几何类型会被设置为 geodb_table。几何类型需要改为 geodb_point，这样就能保存这些点要素。在 Feature Type Properties 对话框中的 General 选项卡中更改设置。



然后检查 Writer 和参数设置，确保各项设置都是正确的，正确的设置如下：

Writer Mode: INSERT

Table Exists: No

Drop Existing Table: No

再次运行这个工作空间时，我们要将选项 Drop Existing Table 设置成 Yes，这样就不会读取到重复的要素。

4）保存和运行工作空间

保存工作空间为 Example2.fmw，这样我们就可以多次运行它了。运行这个工作空间，当被提示时，输入 Geodatabase 的名称：C:\FMEData\Resources\UC2009_ESRI\CityData.gdb。用这个新的目标数据集的名字再次保存它。在本课程后面，我们将会看到更新数据的方法。

使用 Geodatabase 的 writer 来创建域表（domain tables）和子类（subtype classes）。在将要写入的属性中它们被定义为一种特别的数据类型，用来构建域或者子类表作为添加到地理数据库中的要素。

例 3 – 格式属性和域（Format Attributes and Domains）。

这个例子里，我们将加载一些水系分布的数据到要素数据集中。我们有一个包含了不同图层的源 DWG 文件。文件中有一些线，其中包含了我们可以创建域表的属性

1) 创建工作空间

创建一个工作空间，加载一些水系分布数据到要素数据集。我们不要加载 water_meters，因为它们是在下面的例子中使用的。

同样因为一些要素是逻辑相关的，所有我们会用 Feature Dataset 来保存它们。这会在目标要素类型设置中的一个参数来指定。源数据集的设置如下：

Source Format Autocad DWG/DXF

Source Dataset C:\FMEData\Data\Water\distribution_N26.dwg

Source Settings:

Group By: Attribute Schema

Expand Blocks into Entities: Yes

Use Block Header Layer: Yes

Resolve Entity Color: Yes

Read Visible Attributes as Text: Yes

Explode MText Entities: Yes

Feature Class: Water_nodes, water_distribution_lines

现在我们忽略了 water_meters 要素类。

目标数据集的设置如下：

Destination Format： ESRI Geodatabase (File-based)

Destination Dataset C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

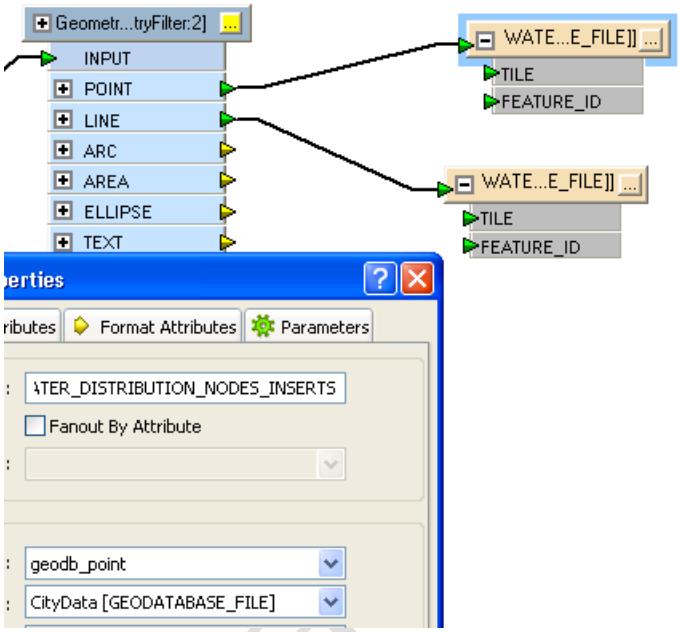
Destination Settings

Feature Dataset: WaterDistribution

Table exists: No

Truncate Table First: No

DropTable First: No

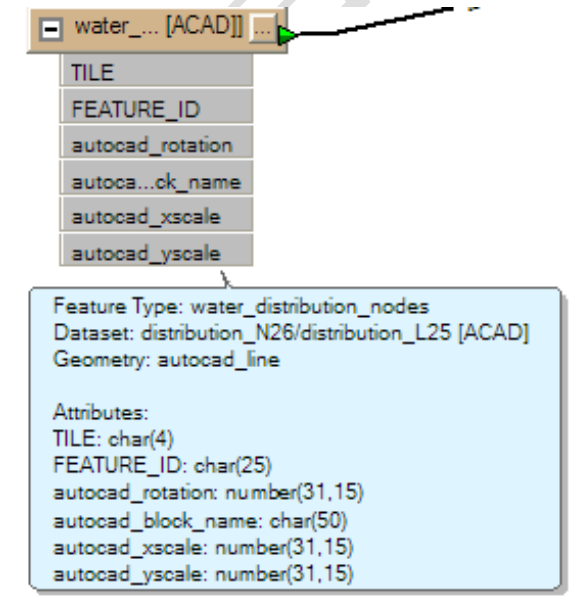


2) 修改目标数据集的名称

分别打开 Water_nodes 和 water_distribution_lines 表的 Feature Type Properties 对话框。把点要素类命名为：water_distribution_nodes_inserts，线要素类命名为：water_distribution_nodes。

3)展开 Reader 上节点的属性

如果我们曾将信息写回到了 DWG 文件，我们就可能需要重建这个块（blocks），这时我们就要暴露 reader 上的必需的属性，将信息放回到块的正确的位置。

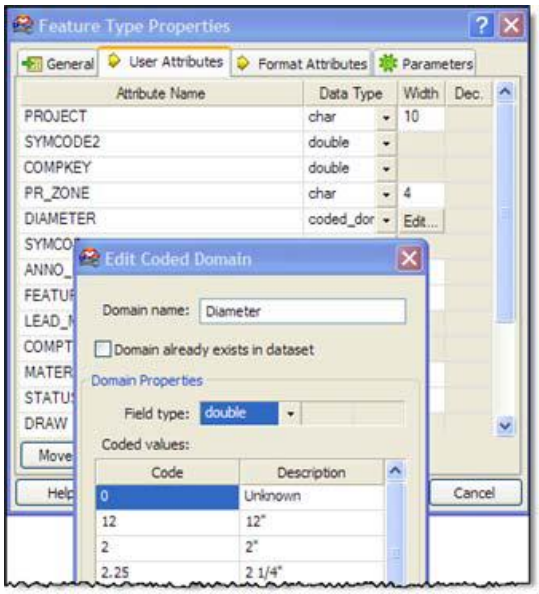


(左图)暴露的属性如下：

- autocad_block_name
- autocad_rotation
- autocad_xscale
- autocad_yscale

4) 建立域表

Water-distribution-lines 数据用来绘制水管，这些水管有一个固定的直径值。我们将为 DIAMETER 属性创建一个域表。打开目标数据集的 Feature Type Properties 对话框，点击 User Attributes 选项卡。找到 DIAMETER 属性，然后点击 Data Type 栏的下拉箭头，找到“coded-domain”选项——选择它，点击 Edit Coded Domain 按钮。



如果在 GeoDatabase 中没有这个域，可以在这里创建它。

不要选中“Domain already exists”复选框。

给这个域表一个名字，如“Diameter”。

选择域的 Field Type 字段，这里设置为“double” 。输入如下代码描述：

0	Unknown
12	12”
2	2”
2.25	2 1/4”
3	3”
4	4”
6	6”
8	8”

5) 保存并且运行工作空间

保存该工作空间为 Example_3.fmw，因为在下面的例子中我们会用到它。

在 ArcCatalog 中浏览要素类，并检查域表（Domain table）。同时检查 WATER_DISTRIBUTION_LINES 和 WATER_DISTRIBUTION_NODES_INSERTS 中的属性，看看域类是如何被处理的，格式属性是如何被保存的。



注意：如果域已经存在，Geodatabase 的 writer 会验证域的定义，如果 Workbench 与 Geodatabase 中的定义不同，就会记入日志（但是不会覆盖原有的域）。

子类型（Subtypes）

子类型允许用户定义表中基于一个域的子分类。例如，一个名为 road 的表中含有一个称为 condition 的域，这个域的值为 good、bad 和 miserable。在 Geodatabase 中，这个域一定要是某种整数类型，这样才能在它上面创建子类型。

在定义子类型（subtypes）时，以下是一些基本准则：

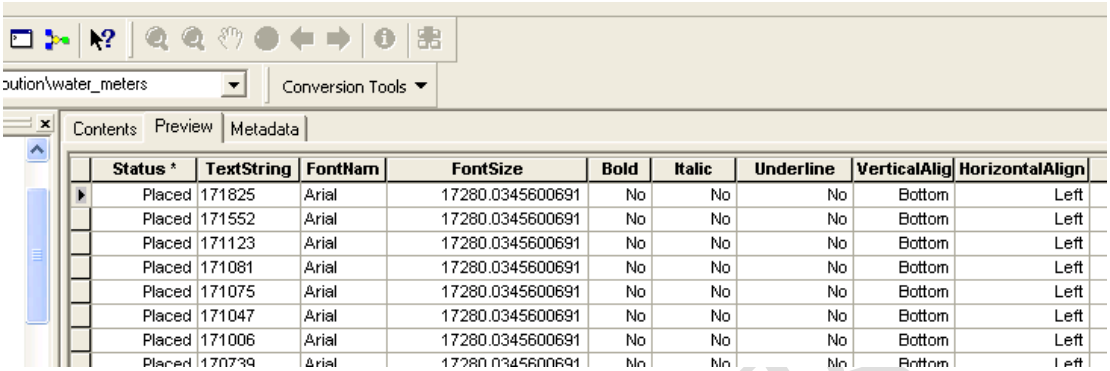
- 每个表中只能有一个子类型。
- 所有的代码必需是唯一的有效整数。
- 所有的代码，说明对必须是唯一的。
- 你不能添加子类型到一个已存在的表中。如果你这样做了，如果子类型存在的话，它会被忽略，表会使用原有的子类型。
- 在写入要素时，子类型的属性一定要包含代码（它被 Geodatabase 作为整数存

储）。

· 想了解更多信息，请阅读 FME Readers and Writers 手册中的 ESRI Geodatabase 一章。

注记提示和技巧（Annotation Tips and Tricks）

在 ArcGIS 9.1 中，注记要素类的结构发生了变化，并且添加了相当数量的额外的字段。这些新的字段包含了注记的相关信息，例如它的字体、大小、角度、偏移、对齐方式等。



Status *	TextString	FontNam	FontSize	Bold	Italic	Underline	VerticalAlig	HorizontalAlign
Placed	171825	Arial	17280.0345600691	No	No	No	Bottom	Left
Placed	171552	Arial	17280.0345600691	No	No	No	Bottom	Left
Placed	171123	Arial	17280.0345600691	No	No	No	Bottom	Left
Placed	171081	Arial	17280.0345600691	No	No	No	Bottom	Left
Placed	171075	Arial	17280.0345600691	No	No	No	Bottom	Left
Placed	171047	Arial	17280.0345600691	No	No	No	Bottom	Left
Placed	171006	Arial	17280.0345600691	No	No	No	Bottom	Left
Placed	170739	Arial	17280.0345600691	No	No	No	Bottom	Left

然而当使用 FME 来读写数据到注记要素类时，这些字段绝不是直接使用；而是必须使用下面列表中的属性来控制注记的属性。完整的表在 FME Readers and Writers 手册中的 Annotation Features 一章中描述。

例如：

属性名称（Attribute Name） 说明（Contents ）

geodb_text_string 这是一个注记内容字串。它被 reader 作为一个 UTF-16 编码字串返回。Writer 转换提供给这个属性的值为 UTF-16 格式。

geodb_text_size 在用户单位中文本的大小。这一大小会被转换成点，当视图在参考比例下时，文本会以这个大小显示。如果没有提供这个属性，就会使用默认的 10 个点大小的文本尺寸，并且没有进行转换。

默认值: 10.0 points

geodb_text_feat_class_name 目标注记要素类的名字。只有在写入要素连接的注记（feature-linked annotations）时，才会用到它。

geodb_text_x_coord 注记位置的 x 坐标。只有在写入要素连接的注记（feature-linked annotations）时，才会用到它。

在使用这些格式属性写入注记时，注释要素类中对应的属性会以正确的值来更新。有些域仍然不被一些格式属性支持，如“Override”。

读入注记（Reading Annotation）

在查看注记时，属性在 ArcCatalog 中与在基础数据表中的显示保存了不同的值。因此，一个属性，例如 Horizontal Alignment，在数据库中显示的是“0”，而在 ArcCatalog 中显示的是“Left”。当用 FME 读取注记，两个值都可看到，也就是说，属性 HorizontalAlignment 显示的是“0”值，而 geodb_h_align 显示的是“Left”值。如果你想以另一个支持对齐的格式写入这个注记，那么弄清楚要使用哪些属性非常重要。



在 Visualizer 里显示要素是非常有用的，因为会显示出所有的属性——包括 ESRI Geodatabase 属性和 FME 格式属性。

分离复合注记（Split Multi-Part Annotations）

在 GeoDatabase 的 reader 中，有一个参数是用来对复合注记进行处理的。在读取数据时，这个参数指定当读入多重注记时，是否要将这个注记分裂成对于每个“元素”(“element”)的单个要素。

如果设置成 YES，在读取数据时，会生成多重注记中的单一要素（通常是一个单词），产生特定的要素属性，例如角度和文本位置，会根据每个要素的位置来保存。如果设置成 No，就会正常读取复合注记，作为存储单个属性集的单一要素。

如果你要读取曲线化的复合注记，这个参数就是非常有用的。这种情况下，注记的每个元素都会有一个属性，这个属性对写入到支持弯曲的注记的格式非常有用。

写入注记（Writing Annotation）

写入注记可能会产生不少困惑，因此我们通过两个写入注记到我们的 Geodatabase 的简单的例子来说明。第一个例子很简单，就是使用从源要素中获取的样式要素来写入注记要素类，第二个例子展示如何加载要素相连的注记。

例 4 – 写入注记

我们将继续使用上个例子保存的工作空间来执行加载水系分布数据集的操作。现在，我们需要处理 water_meters 要素类——从中我们将会创建注记要素类。因为这些要素类与余下的水系分布数据逻辑关联，所有我们也会把它们加到 Feature Dataset 中。如果你没有保存工作空间的话，你可以使用下面的文件：

C:\FMEData\Resources\UC2009_ESRI\Workspaces\example3_complete.fmw 。

1)修改工作空间

在菜单条中选择 Source Data > Import Feature Type Definitions，设置如下所示：

Source Format Autocad DWG/DXF
Source Dataset C:\FMEData\Data\Water\distribution_N26.dwg
Source Settings:
Group By: Attribute Schema
Expand Blocks into Entities: Yes
Use Block Header Layer: Yes
Resolve Entity Color: Yes
Read Visible Attributes as Text: Yes
Explode MText Entities: Yes
Feature Class: Water_meters

2) 复制目标

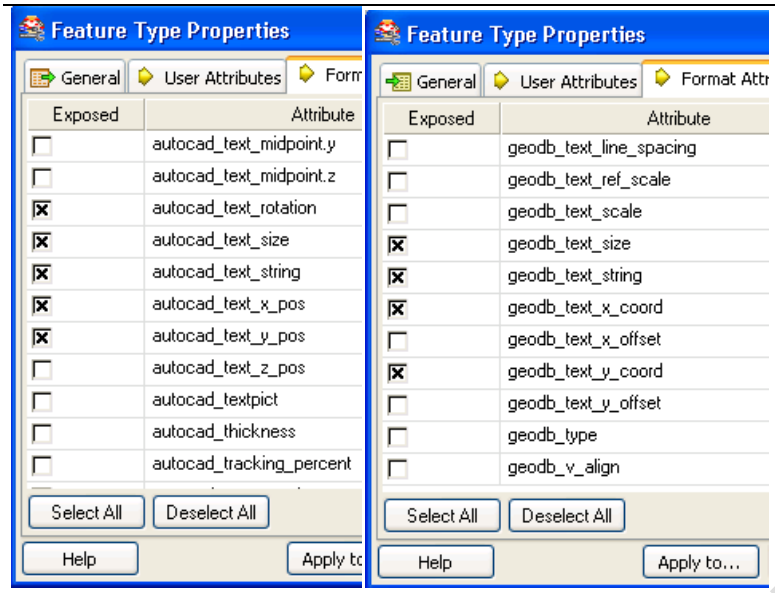
我们需要一个目标数据集来保存注记。在新的数据集源上点击右键，从下拉菜单中选择“Duplicate as Destination”选项。

打开目标要素集的 Type Properties 对话框，编辑 Allowed Geometry 为“geodb_annotation”。

然后点击 User Attributes 选项卡，删除已经添加的属性。我们并不需要指定任何注记属性——它们都是格式属性。点击 OK，保存属性。

3) 暴露源和目标格式属性

在来自 water_distribution_meters 数据源的数据流中，我们需要将不同的文本符号属性暴露到源和目标属性中，以便要素正确的写入。



上图：源数据格式属性和目标数据格式属性

也暴露了下列属性：

-- 在源数据上： fme_color

-- 在目标数据上： geodb_rotation, geodb_font_bold, geodb_font_name 和 geodb_text_color

建立连接的属性和使用 AttributeCopier 替换的连接——提示通过在连接上使用右键菜单可以做到

4) 放置一个 Transformer - AttributeCreator

在 AttributeCopier 后添加 AttributeCreator 来创建属性，给属性赋值如下：

geodb_font_bold – Yes

geodb_font_name – Times New Roman。

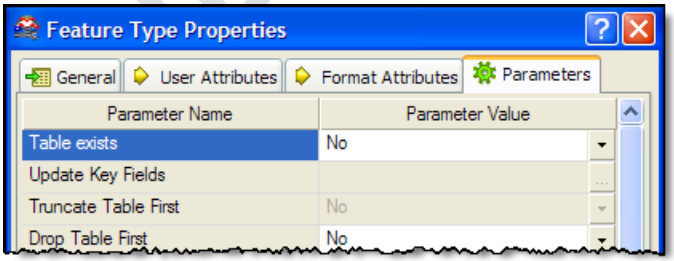
5) 保存和运行工作空间

保存和运行工作空间。

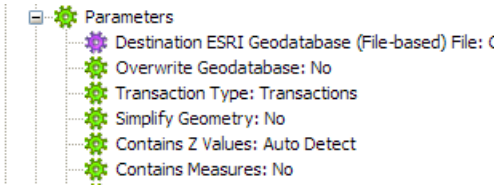
当被提示时，目标空间数据库设置为 C:\FMEData\Resources\UC2009_ESRI\CityData.gdb。

在 ArcCatalog 中查看要素类，检查注记属性值，以及观察要素类怎样在预览窗口中显示的。

检查 Table Exists、Drop Table First 和 Truncate Table 选项的值，然后尝试用不同的组合来重新运行这个工作空间。



不要将“Overwrite Geodatabase”参数设置成 YES，为什么不能这么设置呢？



如果注记要素类已经存在，并且定义了一些注记类，那就有可能要暴露 geodb_anno_class_id 属性，指定使用这个类。一些 geodb_format 属性会在类的说明中重写——在 FME Readers and Writers 手册中查看这方面的更多信息。

要素相连的注记（Feature Linked Annotation ）

正如我们在前面的例子中所看到的一样，注记被存在单独的要素层中，但也可以通过要素连接注记与其他要素进行连接。当注记要素层和其他要素层之间有关联时，就会发生要素连接。在 ArcGIS9.3 中，当创建了要素层时，就会进行要素连接。在 ArcGIS9.3 之前，有必要通过属性 geodb_linked_feature_id 告知 FME 进行要素连接



注意：如果在数据转换之前，已经创建了必要的要素类和关联，就能够使用 FME 读取和写入要素关联注记（Feature-linked annotations）。目前，只能读取或插入要素连接注记，而不能更新或是删除它们。

如果与要素连接的注记还没有写入，那么 Geodatabase 的 writer 就可能写入要素，检索新要素的对象的 ID，接着写入连接它的注记要素。结果是一个 FME 要素包含足够多的信息来写入两个要素：一个注记要素和一个非注记要素。

让我们在一个工作空间来完成这一操作。

我们有一些含有保存了地块 ID 的点质心的宗地划线（property parcel linework）。我们将会加载一个地块的多边形要素类和地块相关的地块 ID 的注记要素类。



例 5 – 加载要素连接的注记

创造一个工作空间来加载数据到一个要素数据集中。我们想要一个关联连接注记（这些注记指定了地块号）的每个地块的要素。我们已经有了属于保存地块的 ID 的质心的界址线（property linework）。为了写入要素连接的注记，要素类必须存在并且定义了注记连接。使用 ArcCatalog 来检查空间数据库，以确保已经进行了加载。

1) 创建一个空白工作空间

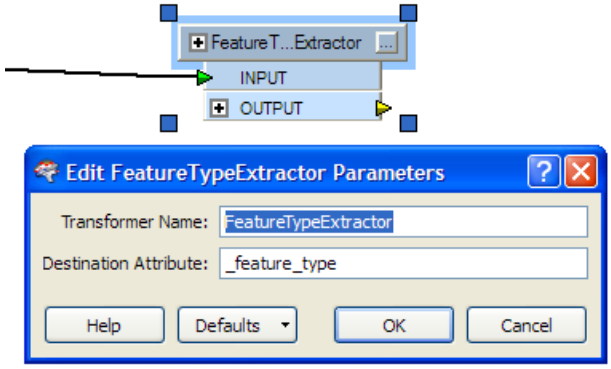
在菜单条中选择 Source Data > Add Dataset，添加一个 MIF/MID 格式的源数据集。设置如下：

Source Format MapInfo MIF/MID

Source Dataset C:\FMEData\Data\Properties\parcel_N26.mif

2) 放置一个 Transformer——FeatureTypeExtractor

我们需要一个所有要素的要素类型名称，这样就可以设置 CITY_GRID_ID 字段。添加一个 FeatureTypeExtractor 到工作空间，使用这个属性的默认名。



3) 放置一个 Transformer——GeometryFilter

添加几何过滤器将线要素和点要素分开。点要素含有一些我们可以来建立注记的属性。对于线要素我们将建成多边形，附上一个城市网格标识符来写出。

4) 添加一个目标要素集

在菜单栏中选择 Destination Data > Add Dataset 选项，添加一个目标 Geodatabase。设置如下：

Destination Format: ESRI Geodatabase (File-based)

Destination Dataset: C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

当被提示添加一个要素类时，在 Properties 对话框中将 Feature Type 命名为 Parcels，几何类型设置为 geodb_polygon。检查目标参数是否如下设置：

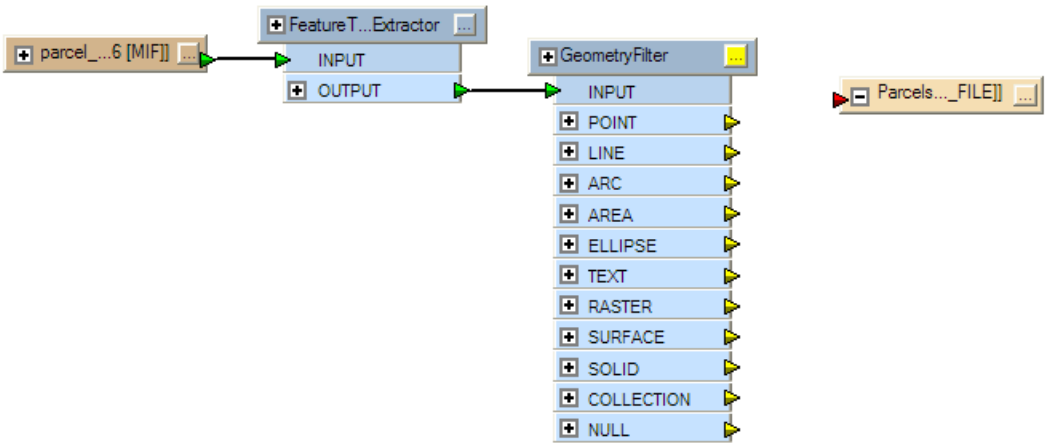
Destination Settings

Table exists: Yes

Truncate Table First: No

DropTable First: No

工作空间看起来应该如下图所示：



5) 放置一个 Transformer——AreaBuilder

添加一个 AreaBuilderTransformer，从 GeometryFilter 的 LINE 端口的输出来得到我们组成多边形的界址线数据。我们想通过片数来定义宗地，所以让我们把选项“Group By” 设置为 fme_feature_type 。

6) 放置一个 Transformer——PointOnAreaOverlayer

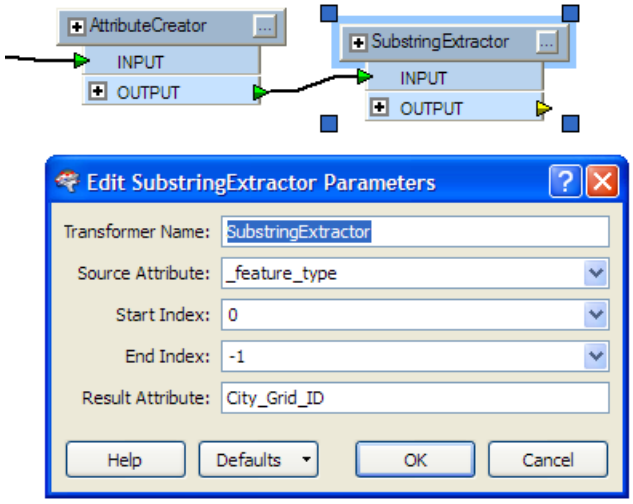
这个 PointOnAreaOverlayer 能转化点的属性到它周边的面上。在面板中放置一个 PointOnAreaOverlayer，然后将 GeometryFilter 中的‘POINT’端口和 AreaBuilder 的‘AREA’

分别连接到 PointOnAreaOverlayer 的‘POINT’和‘AREA’端口上。对这个 Transformer 我们不需要进行任何特别设置。

7) 放置 Tranformers——AttributeCreator 和 SubstringExtractor

操作 PointOnAreaOverlayer 的‘AREA’端口，我们将给每个多边形添加一个属性，这些多边形含有从 fme_feature_type 属性中抽取的 City_Grid_ID 属性。

使用一个 AttributeCreator 来创建 City_Grid_ID 属性——不填写这个值，然后放置一个 SubstringExtractor 来提取 ID 号码。



8)连接目标数据集

连接到目标数据集。记住一定要检查 Destination Dataset Feature Type 属性设置。我们再次更改了要素的几何类型，因此需要确保目标几何设置为 geodb_polygon。同时检查下列参数的值：

Table Exists: Yes

Drop Table: No

Truncate Table: No

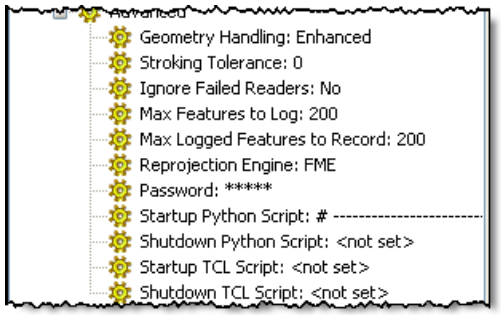
9）保存和运行工作空间

保存工作空间为 Example5.fmw 并且运行它。

10) 重新运行工作空间

尝试重新运行这个工作空间会产生一个有趣的问题。虽然我们设置了 Truncate Table 成 YES，然而只会清空多边形层，不会影响到注记层。因此，要清空这两个层的要素，就必须要使用启动 Python 脚本，引入一些 geoprocessing 工具来完成这个操作。

在文件 c:\Resources\UC2009_ESRI\Workspaces\Parcels_delete.py 上点击右键，并用 python 编辑器——IDLE 来编辑这个文件。它是一些简单的 python 代码，用来引入 geoprocessing 工具以便能从这两个层中删除要素。复制和粘贴代码到 Advanced 设置下面的控制面板中的 Startup Python script 编辑器上。



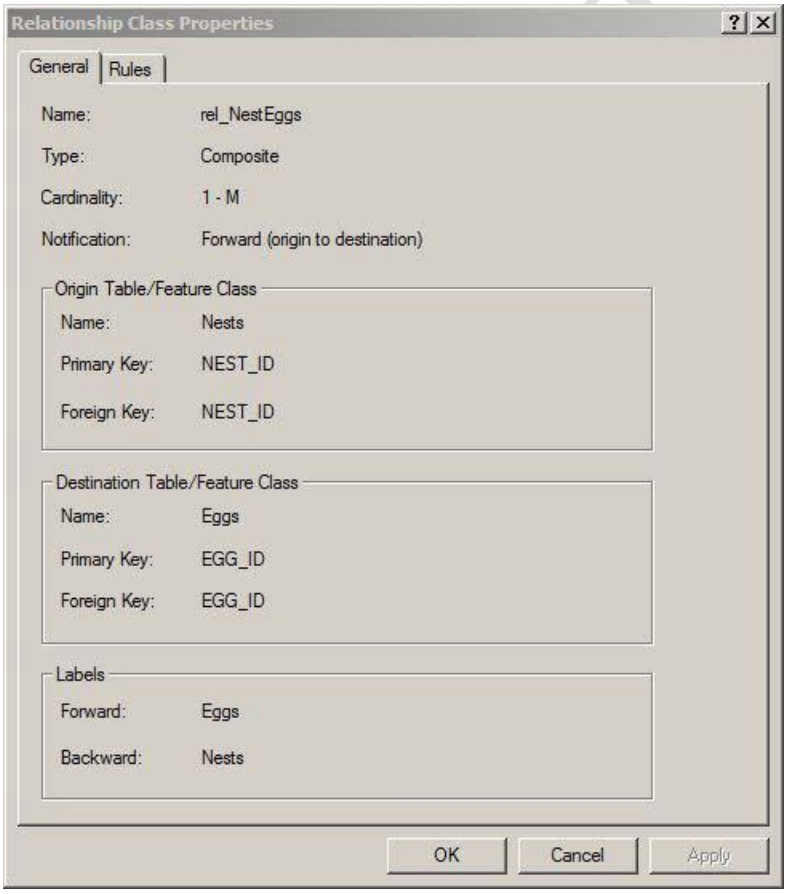
11) 保存和运行工作空间

现在你就可以重新运行这个工作空间，不会创建重复的注记要素。

检查 ArcCatalog 中的结果来确认这一转换工作。

操作关系类

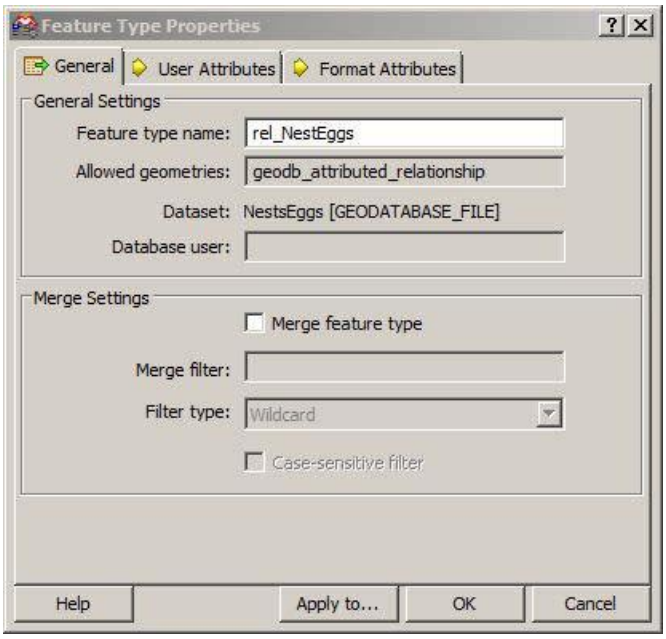
Geodatabase 关系类用于管理一个类中的要素与另一个类中的要素的关系。使用 FME 可以读取和写入属性和非属性关系类。关联并不像其它要素那样是数据表中的一行，而是通过一个源或目标要素的主键和外键值引入的。“Origin”要素属于“Origin Table/Feature Class”，当在 ArcCatalog 中创建关系类时被指定，“destination”要素属于“Destination Table/Feature Class”。



左图：在 ArcCatalog 中一个关系类的属性。“Nests”是原始表（Origin Table），“Eggs”是目标表（Destination Table）。

读取关系类

读取关系类时，必须要同时读取源要素类和目标要素类。



对于关系类的 geodb_type 有两种情况：geodb_attributed_relationship 和 geodb_relationship。

当从关系类中读取时，下列属性被保存在每个关系类中：

geodb_rel_origin_oid：关联原始要素的对象 ID（objectid）。

geodb_rel_destination_oid：关联目标要素的对象 ID（objectid）。

在工作空间中使用这些要素连接源要素和目标要素。

注意：从关系类中读取要素是非常慢的，因为需要读取和查询关系中的每个要素。如果不需要使用这项功能，建议通过把“Ignore Relationship Info”设置为 YES 来关闭读入关系类。

写入关系类

不能通过 FME 来创建关系类，必须通过 ArcCatalog 在转换前建立。在写入要素到关系类时，在要素中下面的属性必须被提供：

geodb_rel_origin_oid: 关联源要素的对象 ID（objectid）。

geodb_rel_destination_oid:关联目标要素的对象 ID（objectid）。

geodb_type: geodb_relationship 或是 geodb_attributed_relationship

以下要素必须要保存到所有有关系的源要素类/表和目标要素类中：

geodb_oid: 要素的对象 ID（objectid）（匹配源数据的 geodb_rel_origin_oid 和目标数据的 geodb_rel_destination_oid）

geodb_feature_has_relationships: 设置成“YES”来指定一个要素参与一个作为源或者目标的关系。

注意上面提到的“objected”值不会转移到 Geodatabase 中。它们仅仅被提供，所以 Geodatabase 的 writer 知道哪些要素是有关系的。

写入关系类的所需的属性如下：

Object	Required Attributes
原始要素类或表（Origin feature class or table）	geodb_oid geodb_feature_has_relationships = yes

目标要素类或表（Destination feature class or table）	geodb_oid geodb_feature_has_relationships = yes
关系类（Relationship class）	geodb_rel_origin_oid geodb_rel_destination_oid geodb_type = geodb_relationship or geodb_attributed_relationship

下面的例子展示这些 geodb_oid 写入后是怎样的：

Origin feature: geodb_oid= 1

Destination feature: geodb_oid=2

Relationship feature: geodb_rel_origin_oid = 1, geodb_rel_destination_oid = 2



如果在工作空间中你有两个或两个以上的空间数据库的 writer，就要注意了。参与了关系和关系类的源和目标要素类（或表）一定要使用同一个空间数据库的 writer 来写入（也就是，你不能用一个空间数据库 writer 来写入要素类，却用另一个 writer 来写入关系类）。

属性关系可以被插入、更新和删除，而非属性关系却只能插入和删除。属性关联有一个与它们相关的中间表，这个表可以通过提供一个 RID（关系 ID）作为关键字来更新，这和在更新表或要素类时，一定要提供 OBJECTID 很相似。



例 6 ——写入到一个已有的关系类中

Interopolis 市的鸟类协会有一个鸟窝的 GIS 数据集和鸟蛋记录的 CSV 数据库。他们想迁移这个数据到空间数据库中，填写一个已存在的 Nests 点要素类、鸟蛋记录的表和存储了鸟窝和鸟蛋关系的关系类中。

1) 启动 Workbench

启动 Workbench，打开提供的工作空间。

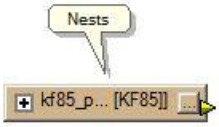
你可以从下列目录中找到这个工作空间：

C:\FMEData\Resources\UC2009_ESRI\Workspaces\example6_start.fmw

这个工作空间中含有源数据集。

选择 File > Save 菜单将工作空间保存到你选择的位置。

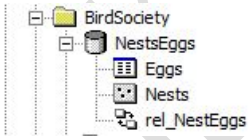


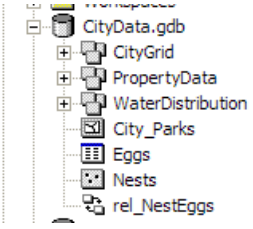


2) 检查地理数据库

在 ArcCatalog 中检查 Geodatabase 文件：

C:\FMEData\Resources\UC2009_ESRI\CityData.gdb。





3) 创建目标数据集和要素类

选择 Destination Data > Add Dataset，添加 CityData 文件 Geodatabase 到你的工作空间。当被提示是否要添加新的要素类到数据集中时，选择“NO”。

选择 Destination Data > Import Feature Type Definitions 菜单，添加 Eggs 表、rel_NestEggs 关系类和 Nests 要素类到工作空间。



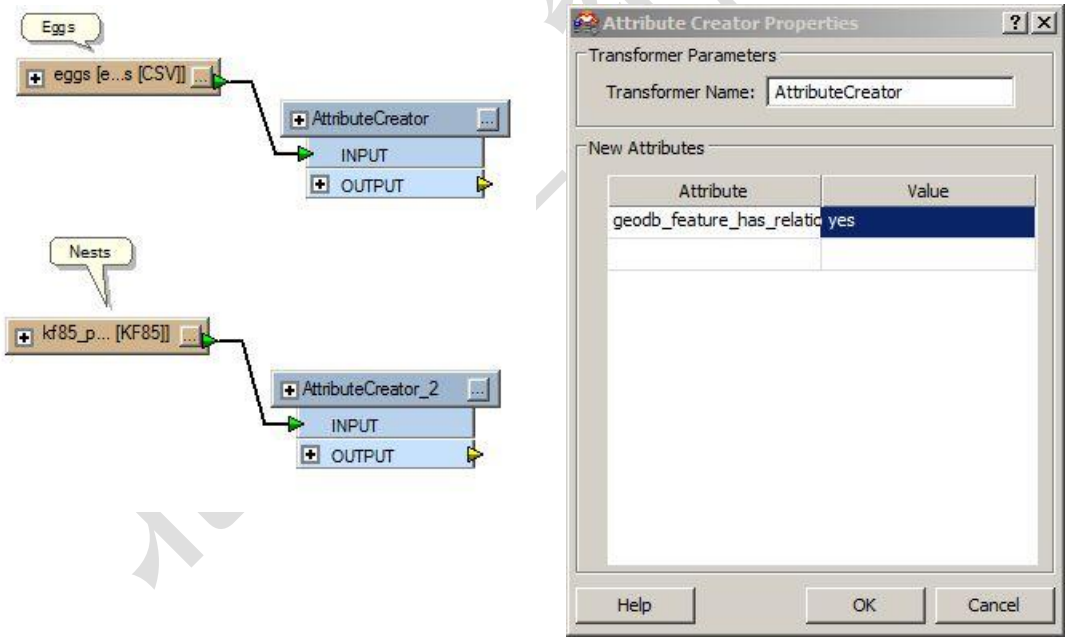
4) 调整 Writer 的参数——Transaction Type

关系类必须被写入到一个编辑模式。更改 Geodatabase 的 writer 的 Transaction Type 为 Edit Session (Parameters > Transaction Type)。



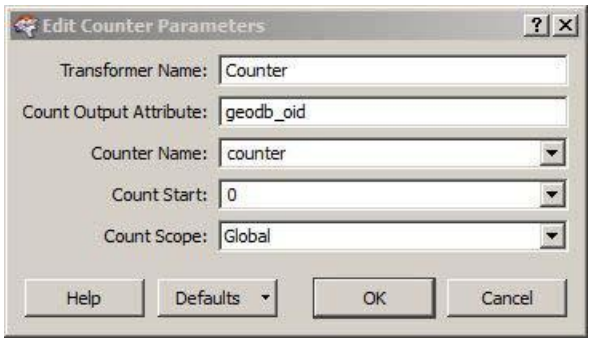
5) 放置一个 Transformer—— AttributeCreator

为 Eggs 和 Nests 要素类都创建一个需要的关系属性 geodb_feature_has_relationships。在每个源要素类后都放置一个 AttributeCreator，设置 geodb_feature_has_relationships 属性值为 YES。



6) 放置一个 Transformer – Counter

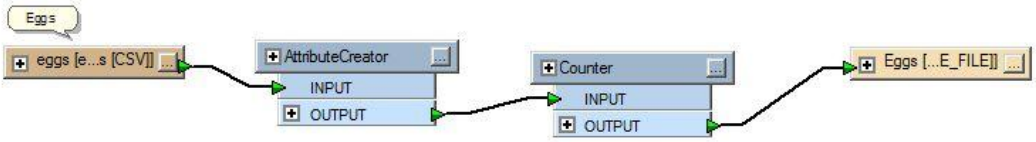
在参加关系类（Eggs 和 Nests）的要素中一定要有唯一的 id (geodb_oid)。当写入这个关系类时，要参考这些 id。在每个 AttributeCreator 后面放置一个 Counter，设置选项 count output attribute 为 geodb_oid。



右图: Counter 的设置。

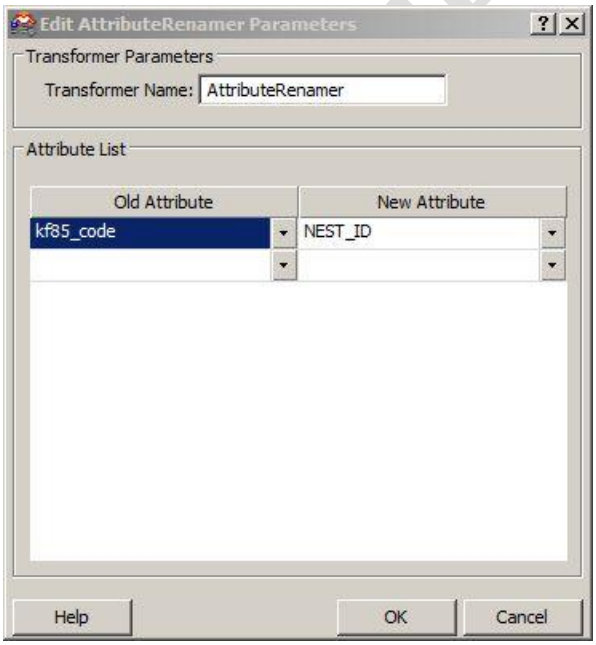
7) 连接到目标要素类——Eggs

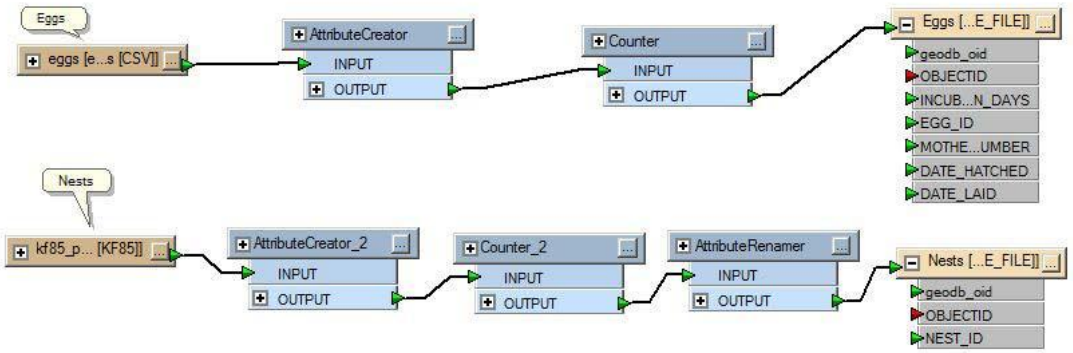
把鸟蛋要素写入到 Eggs 目标要素类中，不需要进行其他的设置。将 Counter 中的输出端口连接到 Eggs 目标数据集上。



8) 放置一个 Transformer——AttributeRenamer

在把鸟窝要素写入到 Nests Geodatabase 要素类之前，必须要进行最后的转换。属性 kf85_code 必须要映射到目标要素表结构的 NEST-ID 上。使用 AttributeRenamer 来进行这一操作。





9) 运行工作空间

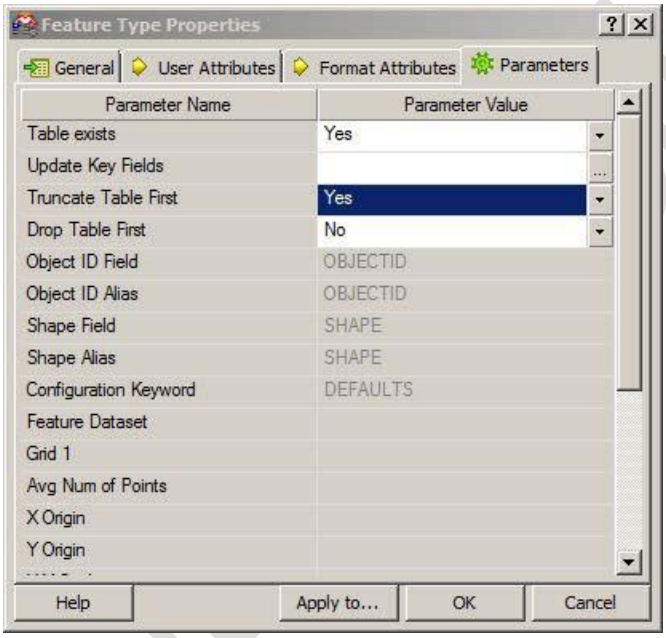
保存和运行这个工作空间。

10) 检查输出

在 ArcCatalog 中检查 Eggs 表的 1507 项纪录和 Nests 表中的 312 个点。

11) 调整目标的 Feature Type Parameters 对话框 – Truncate Table First

确保下次再运行工作空间时，已经首先清空了 Eggs 和 Nests 表。在目标的 feature type properties 对话框中找到选项 Truncate Table First，然后把它设置成 YES。



12) 放置一个 Transformer——AttributeRenamer

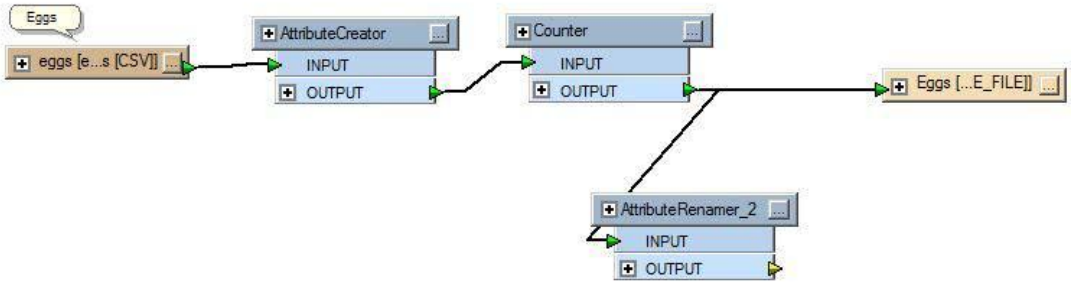
被写入到关系类 rel_NestEggs 的要素必须要同时有两个属性：eodb_grel_origin_oid 和 geodb_rel_destination_oid。这些属性值一定要与写入到源要素类和目标要素类中的 geodb_oid 值相匹配。

为 Eggs 要素设置 geodb_rel_origin_oid 或 geodb_rel_destination_oid 属性。在 Counter 后使用 AttributeRenamer 将 geodb_oid 属性重新命名为 geodb_rel_origin_oid 或 geodb_rel_destination_oid

注意！如果你不是很确定要设置哪一个属性，就在 ArcCatalog 中检查下 rel_NestEggs 的要素类属性，看看 Eggs 是源属性或目标属性。

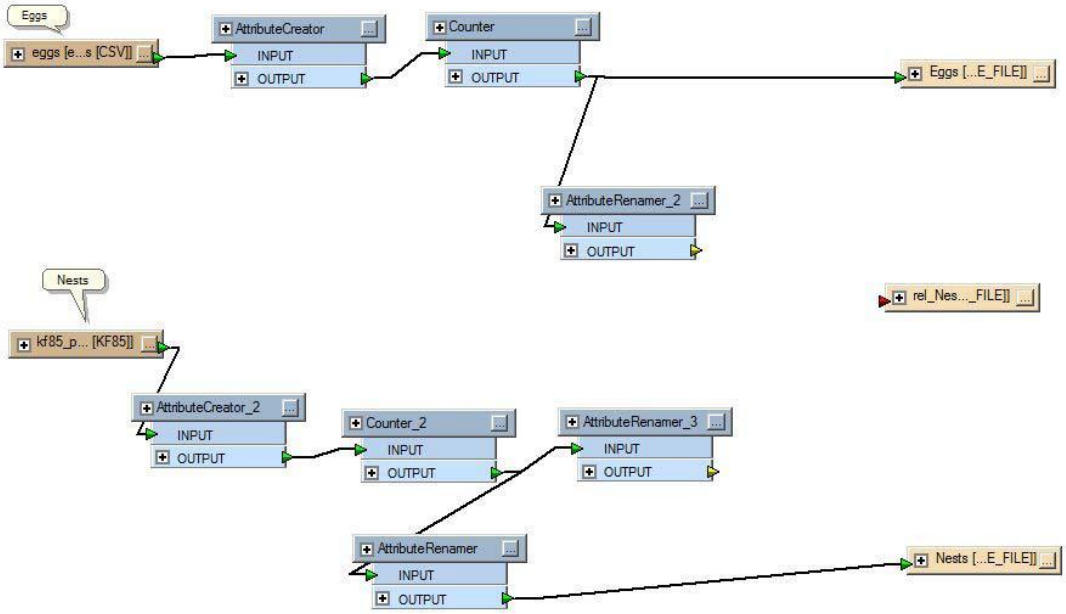
你会将这个 transformer 放置到哪里呢？记住我们将要创建一个转换来写入要素到关系类中，因此我们最终需要合并 egg 和 nest 要素，这样才能使每个要素同时有

geodb_rel_origin_oid 和 geodb_rel_destination_oid 两个属性（第 13 步）。你要平行地放置 AttributeRenamer（创建一个新的连接线路）。



13) 放置一个 Transformer——AttributeRenamer

现在我们对 Nest 要素执行同样的操作。在 Nests 转换中的 Counter 后平行地创建一个 AttributeRenamer。同样将 geodb_oid 命名为 geodb_rel_destination_id 或 geodb_rel_origin_oid。



14) 放置一个 Transformer—— FeatureMerger

现在，eggs 要素有一个 geodb_rel_destination_oid 属性，nest 要素有一个 geodb_rel_origin_oid 属性。写入到关系类中的要素一定要同时有这两个属性，因此使用 FeatureMerger 来连接 nest 要素和 egg 要素的属性。

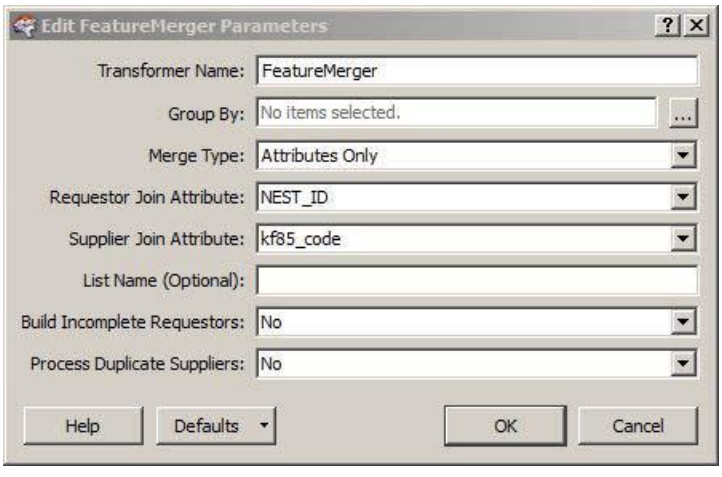
对于连接到 FeatureMerger 的 Requestor 端口的 Egg 要素的连接路径是通过连接 AttributeRenamer 的输出端口到 FeatureMerger 的 Requestor 端口。Nest 要素的路径是从另一个 AttributeRenamer 的输出端口到 FeatureMerger 的 Supplier 端口。

鸟蛋的 NEST_ID 连接到鸟窝的 kf85_code 上。每个鸟蛋只属于一个鸟窝。在 FeatureMerger 属性中设置以下参数：

右图： FeatureMerger 设置

Merge Type: Attributes Only

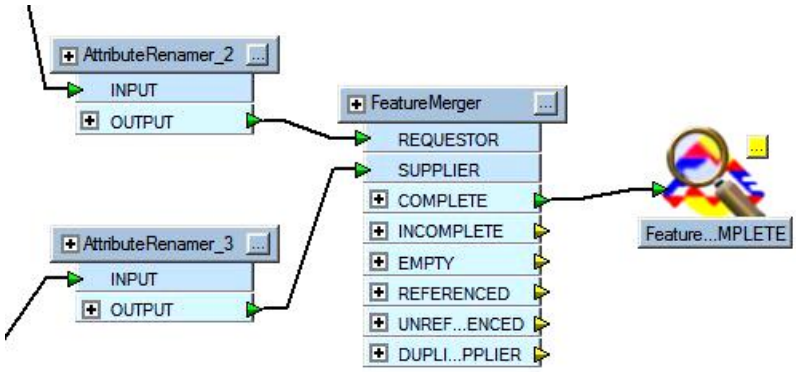
Requestor Join Attribute：INEST_ID



Supplier Join Attribute: kf85_code

15）放置一个 Transformer——Visualizer

现在是检查写入到关系类中的数据的时候。连接一个 Visualizer 到 FeatureMerger 的 COMPLETE 端口（因为我们只对含有相关 nest 要素的 egg 要素有兴趣）。运行工作空间，核查已经添加了 geodb_rel_origin_oid 和 geodb_rel_destination_oid 到一些要素上。记住因为数据是非空间格式，所以你必须选择 No Geometry 工具来查看要素。



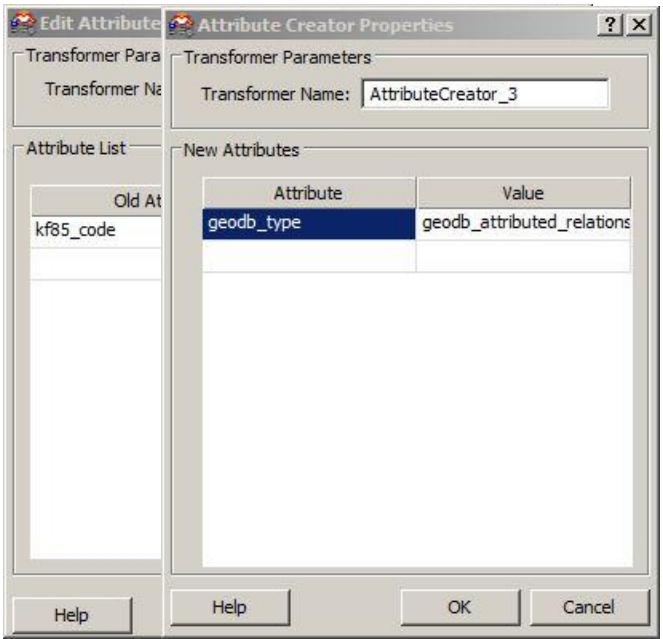
Feature: 1 of 1507
Feature Type: FeatureMerger_COMPLETE
Coord Sys: Unknown

Attribute Name	Attribute Value
csv_type	csv_none
DATE_HATCHED	20040822
DATE_LAID	20040815
EGG_ID	1
fme_basename	eggs
fme_dataset	C:\FMEData\Data\Bir
fme_feature_type	eggs
fme_geometry	fme_undefined
fme_type	fme_no_geom
geodb_feature_has_relationships	yes
geodb_rel_destination_oid	0
geodb_rel_origin_oid	1790
INCUBATION_DAYS	7
kf85_code	303
kf85_internal_name	
kf85_mean_error_height	
kf85_mean_error_plane	
kf85_record_type	Q
kf85_relative_x_coord	3143976
kf85_relative_y_coord	10088891
kf85_rotation	269.01

16) 放置一个 Transformer——AttributeRenamer

kf85_code 需要映射到 NEST_ID。在面板上放置一个 AttributeRenamer，连接 FeatureMerger 的 COMPLETE 端口到 AttributeRenamer 的 INPUT 端口。重命名 kf85_cod 为 NEST_ID。

17) 放置一个 Transformer——

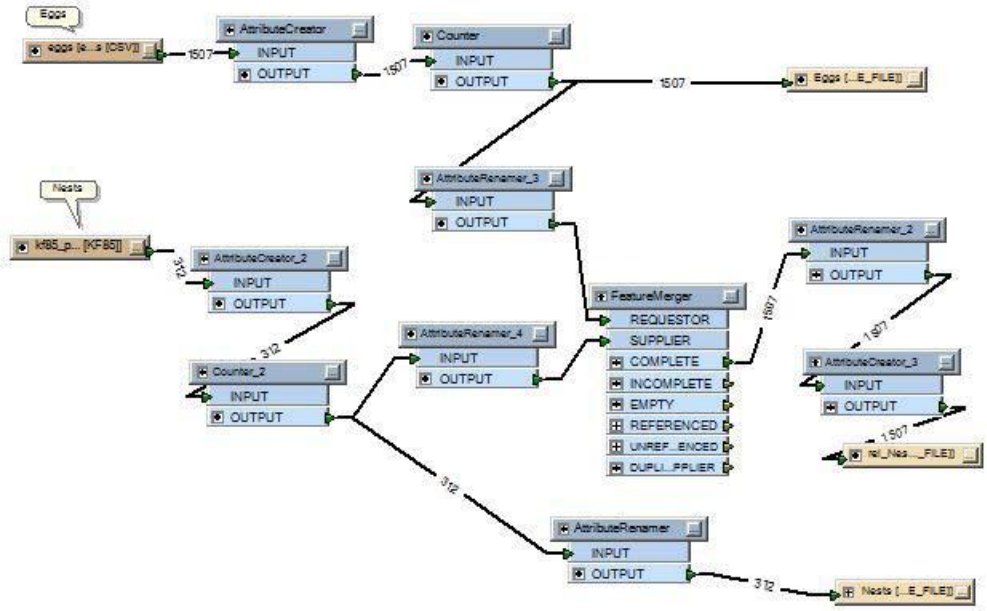


AttributeCreator

现在，为关系类创建最后一个所需的属性：geodb_type。将它的值设置成 geodb_attributed_relationship.

18) 进行最后的连接

连接 AttributeCreator 的 OUTPUT 端口到 rel_NestEggs 要素类。你最终的工作空间看起来是这样的：



19) 保存和运行工作空间

在 ArcCatalog 中检查结果来确保转换的正确运行。

读取和写入几何网络

什么是几何网络？

Geodatabase 的几何网络存储了要素数据集中点和线要素类之间的拓扑关系。它们协助执行数据的完整和要素间的连通性，一般用于公共事业和水文数据。

写入到几何网络

目前，我们不能使用 FME 来创建几何网络或是其中的要素类。在转换数据之前，我们使用 ArcCatalog 来创建几何网络和其中的要素类。然后，我们可以使用 FME 用点和线数据来填写这些存在的要素类，或是作为后期处理，使用创建的几何网络来创建简单的点和线要素类。

当通过设置 geodb_ancillary_role 格式属性写入到一个网络时，指定哪些要素是接收器（sinks）或者源是可能的。可能的值是 none，source 或 sink。

从几何网络中读取

从 Geodatabase 的网络中读取时，许多格式属性（大多都是储存连通性信息的）都会被读出。例如，在读取一个简单边要素时，下列这些属性就会被读出：

geodb_element_id：联接点的逻辑网络要素的 ID；

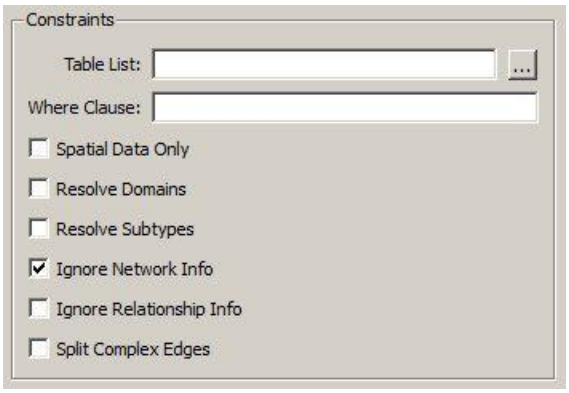
geodb_from_junction_element_id：相当于来自于的端点的连接点要素的 ID。

geodb_to_junction_element_id: 相当于连接到的端点的连接点要素的 ID

查看 FME Readers and Writers 手册了解简单联接点和复杂边要素的可获得的格式化属性。

改善性能

当从一个几何网络中读写时，必须要验证所有的连通性信息。所以，如果忽略网络信息，读写过程就会快很多。为了忽略网络信息，在 Geodatabase 的 reader/writer 的设置中，把选项“Ignore Network Info”旁边的复选框勾选（在工作空间中添加了 Geodatabase 的 reader 后，作为一个 reader 参数，也可以获得这个参数）。



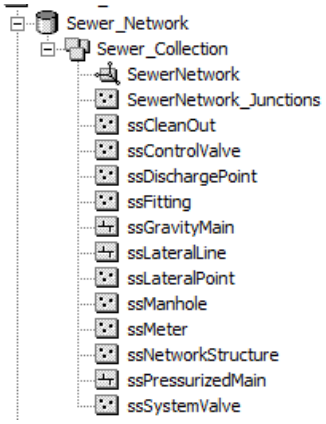
左图：Geodatabase 设置对话框的 Constraints 部分。

例 7 – 加载几何网络

一个城市决定实施一个下水道 Geodatabase 模型，你必须要从 Shape 格式中加载数据到已有的网络要素类。你将会用一个较小的数据集来测试加载的数据，首先使用适当的网络，然后不使用网络，来判断去掉网络，对于完成全部数据的加载是否有价值。

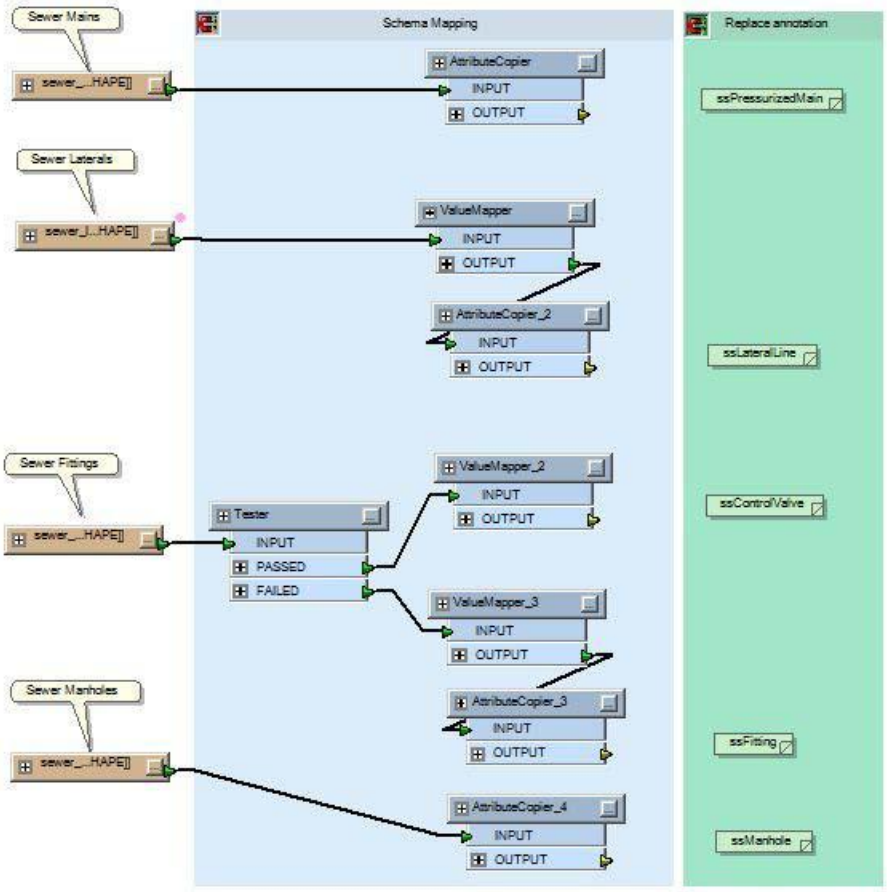
1) 在 ArcCatalog 中检查下水道网络

你将加载下面数据到这个网络：C:\FMEData\Resources\UC2009_ESRI\Sewer_Network.gdb。



2) 打开工作空间

打开工作空间：C:\FMEData\Resources\UC2009_ESRI\Workspaces\example7_start.fmw。这个工作空间已经创建好了，该空间根据转换的需要来加载数据到 Geodatabase 网络。



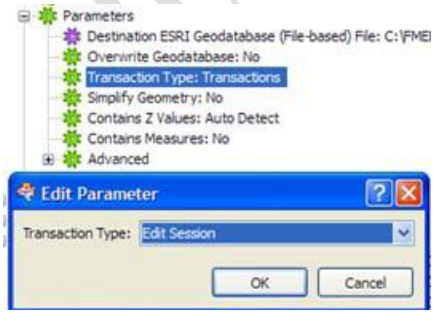
3) 添加目标要素集，导入目标要素类

添加目标数据集和目标要素类来完成了这个工作空间（注释显示了在哪里放置要素类型）。记住你必须写入到已有的要素类，所以在 Geodatabase 的 writer 添加到工作空间中后，必须使用 Destination Dataset 菜单中的选项“Import Feature Type Definitions”。

注意要素类支持的几何属性是： geodb_complex_edge,geodb_simple_edge, geodb_simple_junction 和 geodb_complex_junction。

4) 更改 Transaction Type 参数的值为 Edit Session

因为我们要操作一个复杂的几何类型，更改 writer 上的 Transaction Type 为 Edit Session 是必要的。

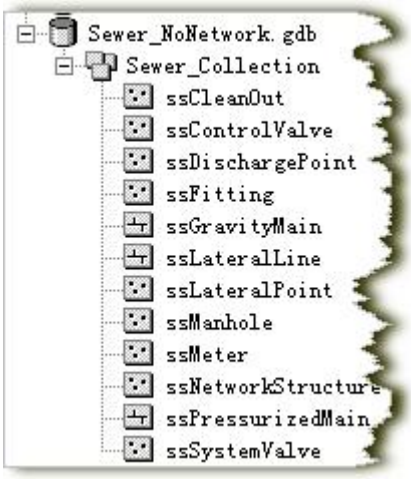


5) 保存和运行工作空间

记录加载数据花费的时间。

6) 浏览 Geodatabase

浏览这个数据库：C:\FMEData\Resources\UC2009_ESRI\Sewer_NoNetwork.gdb



这个 Geodatabase 是刚才被加载的数据库的副本，但是它没有几何网络。

7) 更改目标数据集

更改 writer 参数为“Destination ESRI Geodatabase (File-based) File ”来写入数据到 C:\FMEData\Resources\UC2009_ESRI\Sewer_NoNetwork.gdb

8) 运行工作空间

将运行这个工作空间的时间与上次的作对比。这个要快很多。如果说性能是一个问题的话，如果可能，推荐删掉网络，然后加载完数据后重新创建。

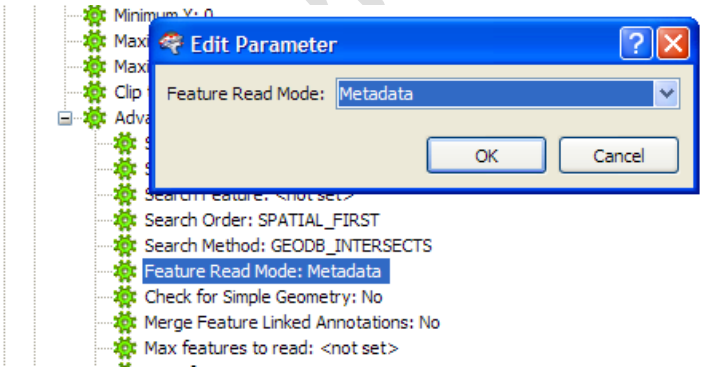
元数据（Metadata）

什么是元数据？

元数据记录是信息文件，通常表现为一个 XML 文档，它获取数据的基本描述或信息源。它代表的是——什么人、是什么、什么时候、在哪里、为什么和资源是怎么样的（who, what, when, where, why and how of the resource）。地理空间元数据用于归档地理数字资源，如地理信息系统（GIS）文件、地理空间数据库和地球影像。一个地理空间元数据记录包括核心数据库目录要素，例如标题（Title），摘要（Abstract），和发布数据（Publication Data）；地理元素，例如地理范围（Geographic Extent ）和投影信息（Projection Information）；数据库元素，例如属性标注定义（Attribute Label Definitions）和属性域值（Attribute Domain Values）。

现在，FME 支持读写 Geodatabase 元数据。

为了能使用 Geodatabase 元数据，你需要设置 Geodatabase 的高级 reader 参数 'FeatureReadMode'为'Metadata'。这个参数引起 reader 读入一个元数据记录，而不是所有常规要素类的记录。



然后，你需要暴露格式属性 geodb_metadata_string 到 reader 上，这样就能在工作空间中读取元数据字符串。这意味着读入要素和元数据，你需要两个 Geodatabase 的 reader:一个用来获取要素，另一个用来获取元数据。

写入元数据需要将要写入的要素写入到一个设置了 geodb_metadata_string 属性的 writer 中，并且将 geodb_type 设置为 geodb_metadata。然而，目标要素类型并不是 geodb_metadata，而是数据表自己的类型（点，多线等）。如果你仅仅要读/写元数据，这些字段会被自动设置。

但是，你要确定暴露和明确的设置它们总是最安全的。如果你需要诊断问题，你也可以查看字段值。

例 8 – 操作元数据

总的来说，FME 不支持内部 writer/reader 的任何元数据解析或字段自动的填写；它只读写 xml 文件作为一个文本块。

在这个例子中，你将从要素类中读取已有的元数据，通过一个 LOGGER 函数来展示它，并且写入它到个人 Geodatabase 中。

1)创建一个空白工作空间

在菜单条中选择 Source Data > Add Dataset，设置如下：

Source Format ESRI Geodatabase (File-based)

Source Dataset C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

Table List: WATER_DISTRIBUTION_LINES

Destination Format ESRI Geodatabase (MDB)

Destination Dataset C:\FMEData\Resources\UC2009_ESRI\CityData.mdb

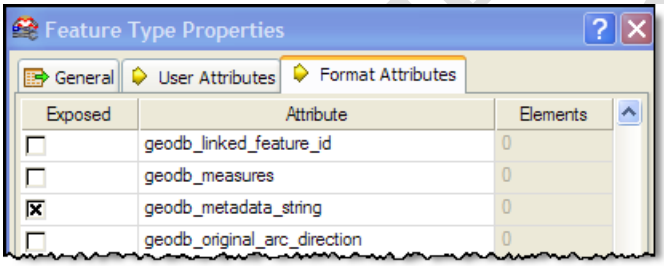
Destination Settings

Table exists: No

FeatureDataset: WaterDistribution

2) 创建 Reader

在导航面板中的 Geodatabase 上的高级参数部分，编辑“Features Reader Mode”参数，将它改成“Metadata”。打开 Source Feature Type 对话框，暴露格式属性 geodb-metadata-string。因为我们只想查看下读入的源数据字符串，所以给这个 reader 添加一个 LOGGER 函数。



3) 运行和保存工作空间

注意在日志窗口中 geodb_metadata_string 的格式。

现在，我们将展示如何写入元数据。

4） 添加第二个 Reader

我们需要添加第二个源数据集来读取真实的要素。选择 Source Data > Add Dataset，添加相同的要素类到工作空间。这次我们将保留 “Features Read Mode” 的设置“Features”。

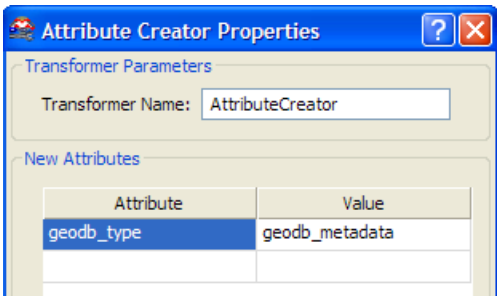
5) 添加一个文本 Reader

在外部文本文件中，我们修改了的元数据 xml 字串，因为我们想用这个新值替换 geodb_metadata_string 的值，所以我们需要读入这个文件。因为我们没有正确的 xfmap 文件来读取数据，所以我们不想作为 XML 文件来读取。我们只是想作为一个单一的包含整个文本字符串的属性，用 Text File reader 来读取它。

选择 Source Data > Add Dataset，添加一个 Text File reader 来读取文件 C:\FMEData\Resources\UC2009_ESRI\Data\metadata_new.xml。你需要更改过滤器，来选择.xml 格式的文件。

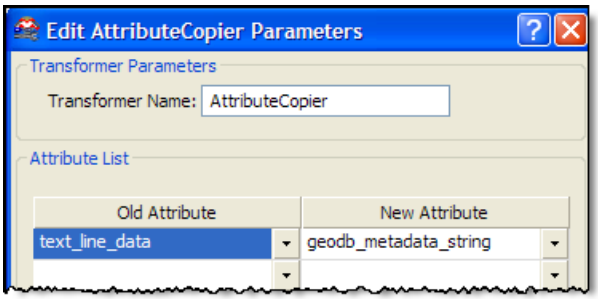
6）放置一个 Transformer——AttributeCreator

在 TextFile reader 后面添加一个 AttributeCreator 函数来增加 geodb_type 属性，并赋值为 geodb_metadata。



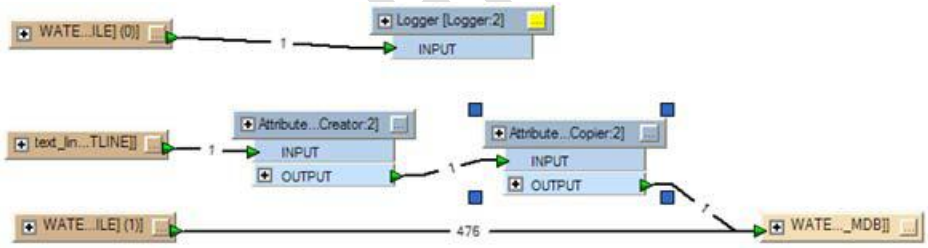
7) 放置一个函数—— AttributeCopier 或 AttributeRenamer

新的元数据作为来自 Text File Reader 的单一块输出，我们需要更改它的名字为 geodb_metadata_string。



8)连接到目标数据集

将元数据要素和源多线要素都连接到目标数据集。注意，目标数据集中有一个多线的 geodb_type 属性，元数据要素将会写入到 writer。工作空间看起来是这样的：



9) 保存和运行工作空间

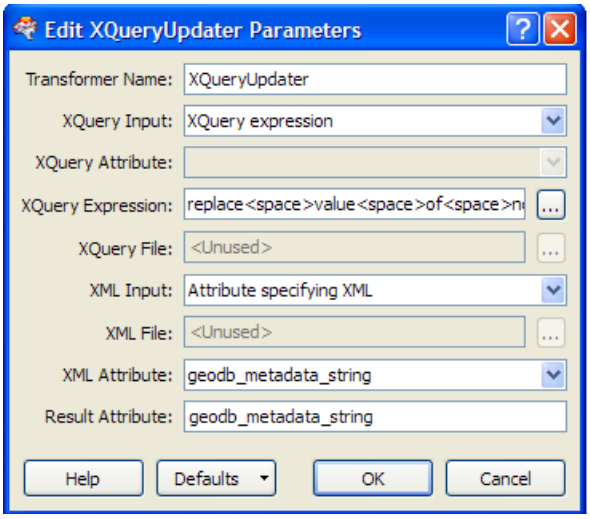
在 ArcCatalog 中检查输出结果。在个人 Geodatabase 的 Water Distribution Lines Feature 类中打开元数据选项卡，就能看到新的元数据。

在前面已经提到了，这只是初步操作 XML 的方法。随着基于数据源的 XML 和用 XML 流响应用户的网络服务的数量的增加，我们开始感到需要使用一个允许从 XML 中用一种简易灵活得多方式来提取数据的工具。FME2009 被 XQuery（一种被设计来查询 XML 数据集合的语言）授权。

对于 XML 的 XQuery 与对于数据库的 SQL 的功能相似。因此对这个工作空间的一个快速的更改来操作将要写入的 XML 是使用一个 XQuery 函数。

10) 放置一个函数—— XQueryUpdater

编辑元数据的另一个选项是利用 XQuery 函数。在元数据的 reader 后放置一个 XQueryUpdater——不是来自 TextFile Reader 的数据流。这将会读入元数据到一个 geodb_metadata_string 属性中。现在修改设置如下：



在 XQuery Expression 选项中使用如下 XQuery 声明：

replace value of node /metadata/idinfo/descript/abstract with "Water distribution line data",

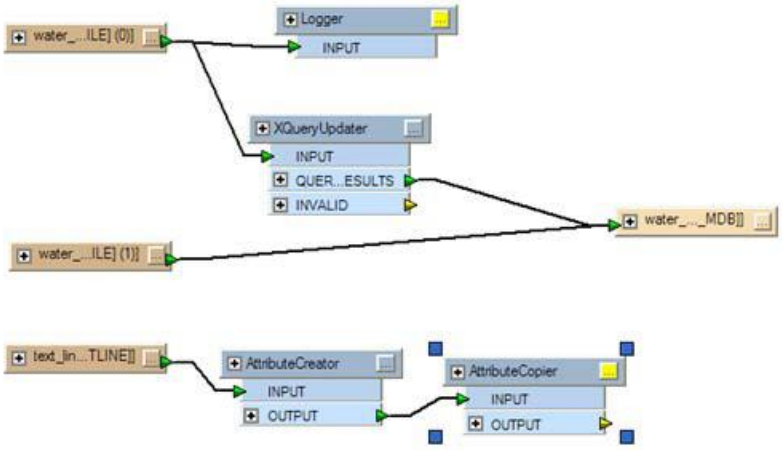
replace value of node /metadata/idinfo/descript/purpose with "Used for maintaining distribution information",

replace value of node /metadata/Esri/CreaDate with "20090520"

11) 保存和运行工作空间

断开与 TextFile reader 的连接或使其无效，然后连接 AttributeRenamer 到目标数据集。

现在工作空间看起来如下：



重新运行工作空间，在 ArcCatalog 中检查的结果。

XQuery 函数提供了一种到 XML 文档和字符串的精密的接口。更多的信息在我们的 fmepedia 的网站上：http://www.fmepedia.com/index.php/XQuery_Examples

更新数据

到目前为止，我们已经做了插入数据到 Geodatabase。如果想更改数据，我们就必须删除或清空要素类，然后重新插入所有的数据。现在我们将查看两个选项来更新已存在的表，这样将不再清空或是删除要素类。

Writer Mode

记得导航面板中 Writer 下的目标数据集的 WRITER MODE 参数吗？这个参数的值是下面

三个中的一个：INSERT、 UPDATE 或 DELETE

INSERT 就是将数据插入到 Geodatabase，它是默认值。不管怎样，如果你用的是已有的 Geodatabase，你可能对更新或删除已有信息更感兴趣。因此，我们来查看 UPDATE 选项，它提供了一种修改 Geodatabase 中的数据的功能。

使用 UPDATE mode 中的 writer，我们有一个选项不仅能更新要素类的信息，如果需要也可以插入或删除这些信息。

Feature Type level 参数是会影响单个要素类的一些参数。在这种情况下的 Geodatabase 每个 Feature Type 代表一个单个表，因此 Feature Type level 参数只影响写入到这个指定数据表中的要素。

例 9 – 更新已有的要素

这个例子会使用到我们之前加载的 AddressPoints 表。现在，我们有一个 CSV 文件来更新数据。每个要读入的要素已经如下标记：U ——updated，已更新要素；D——deleted 要删除的要素，I——insert，插入的要素。我们会使用 UPDATE KEY FIELD 参数来处理这些已更新的要素。

1）创建工作空间

使用源要素的 CSV 文件创建一个工作空间来更新 Geodatabase 中已存在的要素。在工具条中，选择 Source Data > Add Dataset，添加源文件，设置如下：

Source Format Comma Separated Value (CSV)

Dataset C:\FMEData\Data\Addresses\AddressUpdates\AllAddressUpdates.txt

Source Settings Has Field Names = Yes

Lines to Skip = 1

在工作条中，选择 Destination Data > Add Dataset 菜单，添加 Geodatabase 的 Writer。当被提示“add a new feature type”时，点击 NO。我们将更新已存在的要素类，所以需要引入已存在的要素类的结构到我们的工作空间。目标要素类的设置如下：

Destination Format ESRI Geodatabase (File-based)

Destination Dataset C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

然后，在工具条中选择 Data > Import Feature Type >Settings，添加 ADDRESSPOINTS 表，设置如下：

Destination Settings

Feature Class: AddressPoints

Table exists: Yes

Truncate Table First: No

DropTable First: No

Writer Settings:

Writer Mode: UPDATE

这时初始的工作空间就看起来是这样的：



2) 添加 Transformer: 2DPointReplacer

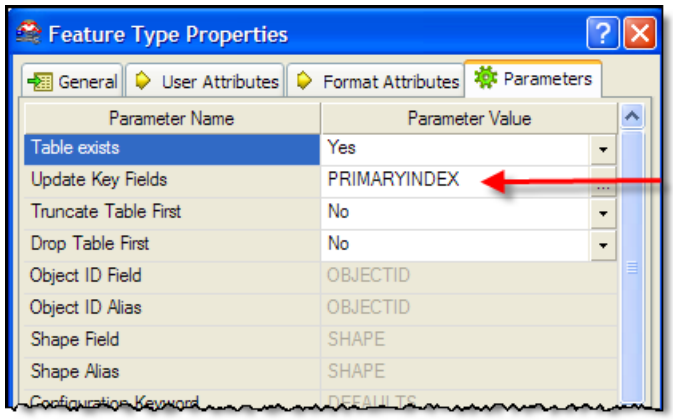
添加一个 2DPointReplacerTransformer 来转换 CSV 记录到真实的空间要素中。X 和 Y 的属性值分别设置为是 X_COORD 和 Y_COORD。

3) 设置 Update Key Field 参数

打开目标的 Feature Type Properties 对话框，点击 Parameters 选项卡。在 Update Key Fields

选项中选择属性 PRIMARYINDEX。

这就意味着这个表中存在的记录将会被将要写入的含有相同的PRIMARYINDEX值的要素所代替。



左图: Feature Type properties 的设置

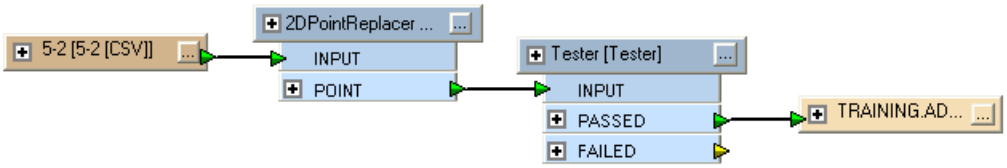
如果 Table Exists 被设置为 NO，就要将它改成 YES。

同时我们也要确保 Truncate Existing Table 设置为 NO。

4) 添加一个 Transformer: Tester

很显然，如果我们已经将记录标志为删除（deletion），我们不希望将它们插入到数据表中，事实上我们只需要保持更新要素，因此，编辑 Tester 转换器的参数，以 UPDATE_TYPE = U 为条件进行测试。

下图: 此时你的工作空间是像这样的。因为我们要一直保留 Tester 的“PASSED”端口的要素，所以测试条件设为 UPDATE_TYPE = U。



5) 添加 Identifier 属性

判断是否在发生了更新可能比较困难，因此在 Tester 后面插入一个 TimeStamper 转换器。用它来设置 timestamp 属性为 LA_UPD_DAT。

添加一个 AttributeCreator (或 AttributeSetter)来设置 LA_UPD_BY 为如你的名字之类的值。

6) 运行工作空间

运行这个工作空间。

使用 FME Viewer 来检查数据，查看是否进行了更新。选择 Features Fileter 选项，来查看哪些要素有 LA_UPD_BY 的值，这个值是你在第五步中设置的。



讨论点：

这个技术的问题是更新、插入和删除作为不同的处理必须分开执行。因为每个要素类(表)

只能在工作空间中表示一次，你不能同时进行所有的操作。

要素级别的更新（Feature Level Updates）

对标签要素使用格式属性

更新数据库和数据表层级别是粗糙的，潜在的要花费大量的时间，因为你可能以写入比需要多的数据，比理想情况更多的处理过程。

使用要素级别（feature-level）可以进行较低层次的更新来控制更新操作，使你的更新更精确。每个要素必须要以某种方式来标注，在 FME 中这使用格式属性来做。

fme_db_operation 属性

fme_db_operation 是一个格式属性，它告诉数据库的 writer 如何处理这个要素。就像 writer 参数，它的值可能是 DELETE、INSERT 或 UPDATE.

fme_where 属性

fme_where 也是一个格式属性，它指定强制一个更新的匹配。例如，一个表达式 MyField = @Value(MyIndex)，是说当数据库字段 MyField 和属性 MyIndex 之间存在一个匹配时要更新要素。

例 10 – 更新已有的数据

在这个例子中，我们准备使用要素级别参数（feature level parameters）来管理数据库，执行对要写入要素的操作。我们会更新已更改的要素，插入新要素，删除标记为不再需要的要素。

1) 使用在例 9 中创建的工作空间

启动在例 9 中创建的工作空间，或使用这个文件：

C:\FMEData\Resources\UC2009_ESRI\Workspaces\Example10_start.fmw

2) 设置 Update Key Fields 参数

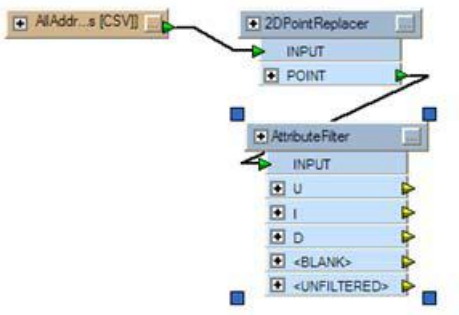
使用 fme_where 的格式属性，你能判断要更新哪些记录。

不过在 fme_db_operation 上设置操作类型是可以理解的，但是要使用要素类的 Update Key Fields 参数来判断查找属性。

所以，就像在例 9 中一样，打开目标的 Feature Type Properties 对话框，点击 Parameters 选项卡。确定将 Update Key Fields 设置为 PRIMARYINDEX。

3) 过滤更新

这次，我们想过滤所有类型的更新，因此，用三个 Testers 分别连接更新类型的三个端口 U、I 和 D。



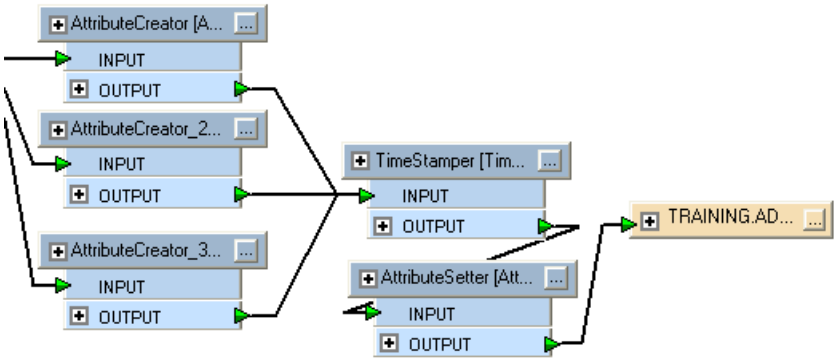
或许使用一个 AttributeFilter，类似于左图。

4) 添加 AttributeCreator 函数

添加三个 AttributeCreator 函数——每个连接 Tester 的 PASSED 输出端口的要素。

使用 AttributeCreators 来创建一个名为 fme_db_operation 的属性。

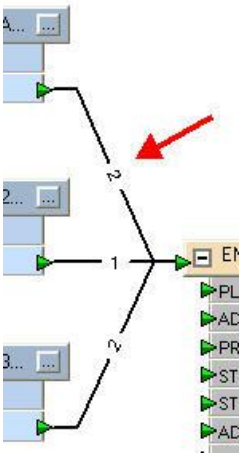
每个转换器应该被设置为不同的属性值 UPDATE、DELETE 或 INSERT——这取决于要处理哪类要素。



这三个 AttributeCreators 都连接到 TimeStamper 转换器上。上图: 工作空间的最后部分应该是这样。

5) 运行工作空间

运行这个工作空间。



每条连接线上的要素计数会告诉你每个表中有多少记录被更新、插入或删除。

在 FME Viewer 中，使用你的用户名就能过滤要素，与源 CSV 文件中所做的更改列表进行匹配。

作为一个额外的任务，你可以试着用最后更新日期（LA_UPD_DAT)来过滤要素，查看刚刚执行的所有更新的要素。

问题：在目标要素类中，可以将 fme_db_operation 设置成常量吗？如果可以，这样做对我们的工作有没有好处呢？

NB: 如果你需要重新运行工作空间（可能是因为你在操作中发生了错误或其它类似原因），那么你就可能要重新储存原始数据表。这仅仅需要重新运行例 2 中的工作空间。

6）进一步的任务

使用 ValueMapper 代替 Tester/AttributeFilter-AttributeCreator，就能使每个要素与他们的 fme_db_operation 相匹配吗？试一下，并找出答案

试着用 fme_where 属性代替 Feature Type properties 参数设置。

变化检测

虽然上面使用的源数据已经被标记成了要求的操作类型，但并不是每个用户都清楚是否已经更改了源数据。在这种情况下，我们就需要进行变化检测。

变化检测

为了检测是否在已有数据和新数据集间作了更改，显然，我们需要同时读取这两种数据，才能进行变化检测处理。比起更新预定义记录，这个检测过程要花费两倍的时间，这是因为这个操作既要读入数据，又要进行变化检测。

变化检测的最大问题就是要删除的记录。我们通常采取的方法是假定新数据中任何不存在的要素都被删除了。不过在这个操作中会让用户处理所有的新数据，因为不这样的话，那些仍在处理的数据也会被假设成已删除的要素，并从数据库中移除。

ChangeDetector 函数顾名思义，ChangeDetector 就是说：它输入两类要素（原始的和改变的），并且对它们进行变化检测。

ChangeDetector 最大的弊端就是，不能检测一个指定项的编辑，只能找出两大类要素间的增加、删除。换句话说——例如使用地址数据库——它不能识别被编辑的特殊记录，只能将它标记成一个新的记录（INSERTION），而把原有记录标记成 DELETION。

了解检测编辑的自定义函数，请登录 ：

http://www.fmepedia.com/index.php/UpdateDetector 。

例 11 – 更新已有的数据

检查文件夹 C:\FMEData\Data\Developer 的内容。

一个开发商打算在城镇的边缘从一个大地块中开发一些小地块。我们想把这些新地块与在这个区域的现有的表进行比较，用这一新信息来更新城市记录。

这样做，你将实现下列目的：

1) 创建一个工作空间来从 Geodatabase 中读取城市当前的地块数据集

2）为开发商的 DXF 数据集添加一个新的数据源的 reader

3）清空所有的源数据，确保更好地匹配

4）使用一个 ChangeDetector 函数来把城市数据和开发商的新的界址线作比较

5）更新现有的 Geodatabase 中的数据

1) 开始工作空间

启动 Workbench，创建一个空白的工作空间。

2) 添加 Reader

在菜单条中选择 Source Data > Add Dataset。在工作空间中添加一个新的 reader，读取地块数据集的设置如下：

Source Format ESRI Geodatabase (File-based)

Source Dataset C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

Source Settings

Table List: Parcels

3) 添加一个源数据集 ——开发商的 DXF 文件

在菜单条中选择 Source Data > Add Dataset。

点击 Browse 按钮，浏览、选择开发商的 DXF 数据集，设置如下：

Source Format Autodesk AutoCAD DWG/DXF

Source Dataset C:\FMEData\Data\Developer\lotlines.dxf

Source Settings

Group Entities By: Attribute Schema

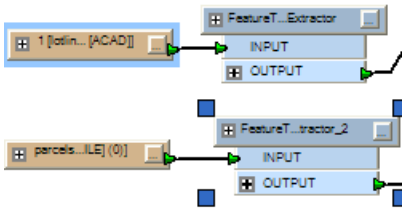
4) 暴露格式属性 - fme_feature_type

现在，我们需要清空源数据，以确保数据集最好的匹配。理想地，通过相同的转换器集合，我们会运行所有的数据。我们也可以平行的操作——也就是说，转换器的两个数据流——但是仅用一个 Transformer 来合并所有的数据，运行这个工作空间会更加有效、整洁。

当我们用这种方法来合并数据，我们需要确保有能够将这些数据分开的方法。源要素类是可以这样做的一个很好的途径，只要我们能在属性中储存它。

格式属性 fme_feature_type 保存了这个信息——因此，一个简单的方法就是在两个源要素类上暴露这个属性。不过 FeatureTypeExtractor 转换器也能执行这个操作。

放置两个 FeatureTypeExtractor 转换器，保持目标属性的默认值_feature_type。一个 FeatureTypeExtractor 连接一个源要素。



右图: 此时的工作空间看起来是这个样子：

5) 放置一个 Transformer—— GeometryFilter

因为我们不能保证开发商是否以某种方式合并或分离了线——即使并没有改变几何要素的形状——所以，我们要将线组成多边形要素。

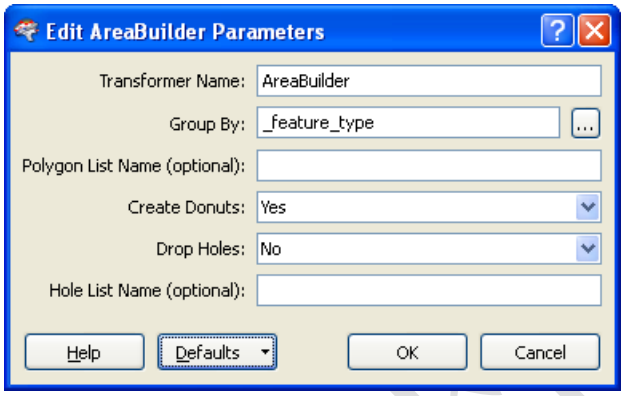
在组成多边形之前，我们要确保仅仅对线要素操作。在尝试建立多边形之前，让我们使用一个 GeometryFilter 来过滤出点要素。

放置一个 GeometryFilter，把它连接到来自开发商的源的 FeatureTypeExtractor 转换器后

面。

6)放置一个 Transformer——AreaBuilder

添加一个 AreaBuilder 转换器，连接它到 GeometryFilter 的 LINE 输出端口上。检查 AreaBuilder 的设置。



设置 Group-By 选项来根据存储在要素类中的分组属性。

如果你使用 FeatureTypeExtractor，要素类就是_feature_type；如果你暴露了一个源格式属性，要素类就是 fme_feature_type。

将 Create Donuts 设置成 NO。

7) 放置一个 Transformer——2D Forcer

因为空间数据库中的数据有可能是 3D 数据，但是我们知道数据没有高程信息，因此需要强制将所有数据作为 2D 数据来处理。将这两个数据流都连接到 2DForcer，而不是复制转换器。

8) 放置一个 Transformer——CoordinateRounder

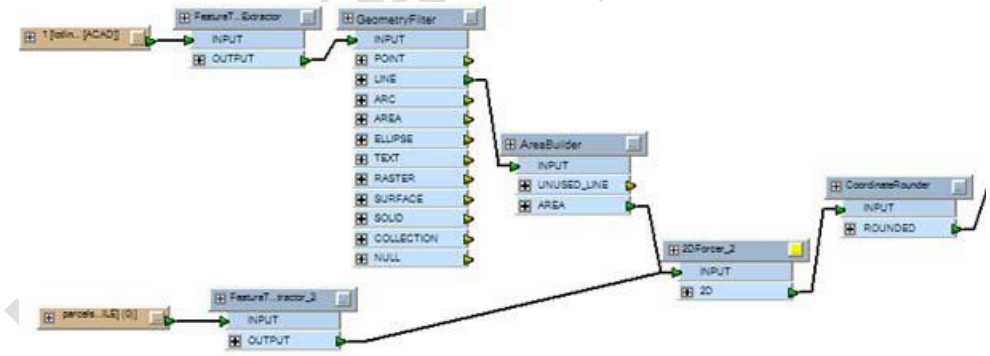
如果开发商的数据精度和城市的标准不同，这时变化检测可能产生不利的影响。为了避免潜在的问题，我们需要添加一个 CoordinateRounder 转换器。

连接它到 2DForcer 的输出端口。

将 X 和 Y 坐标都设置成三个小数位。

这样所有的数据的准确度都是相同的。

这时，工作空间看起来像这样（下图）：



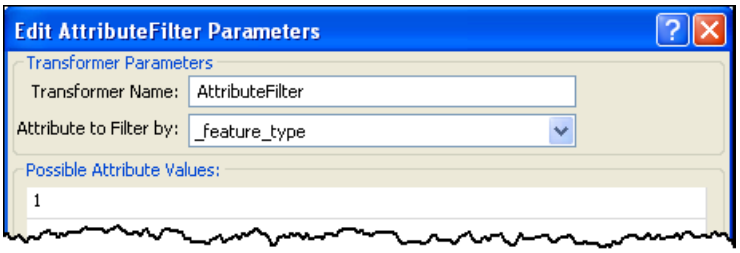
9) 放置一个 Transformer——AttributeFilter

现在是时候将数据重新分成两部分。

这里 FeatureTypeFilter 不能用，因为 AreaBuilder 会清除每个要素的原始的 fme_feature_type 值，所以我们使用一个 AttributeFilter。

放置并连接 AttributeFilter 转换器。

打开设置对话框。选择保存 FME 要素类(_feature_type 或 fme_feature_type)的属性作为过滤的属性。



对于可能的属性值的列表，我们需要做的就是过滤出开发商的 AutoCAD 数据和 city 中的一切数据。

对于开发者的数据，源

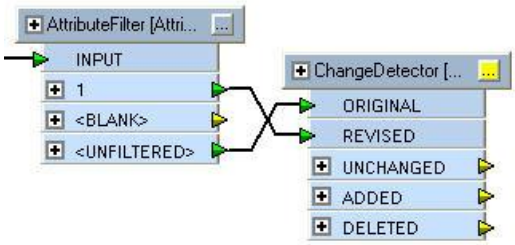
要素类是“1”，因此只要简单的输入数字 1 到值列表,点击 OK。

10) 放置一个 Transformer——ChangeDetector

在 Transformer 面板上找出 ChangeDetector 函数，将它添加到工作空间中。

连接 AttributeFilter 的 1 端口到 ChangeDetector 的 REVISED 端口。

连接 AttributeFilter 的<UNFILTERED> 端口到 ChangeDetector 的 ORIGINAL 端口。



右图: 这部分的工作空间看起来是这样的。

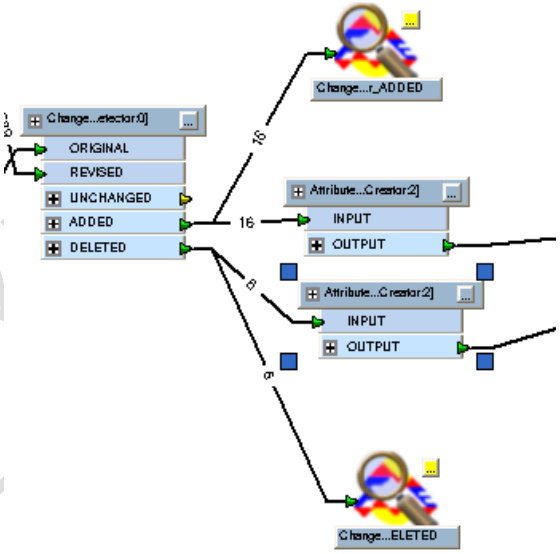
11) 检查 Transformer 的设置

检查 ChangeDetector 的设置，确保 Match Geometry 设置成‘2D’，Lenient Geometry Matching 设置成‘YES’。确定 Attributes to Match 是空的。

如果要素有相同的坐标系，但是——例如有相反的顺序，Lenient Geometry Matching 参数能保证匹配。

12) 放置 Transformer——AttributeCreator

添加两个 AttributeCreator 转换器来连接 CHANGEDETECTOR 的 ADDED 和 DELETED 两个端口。



创建属性 fme_db_operation，将它设置成 INSERT 或 DELETE（取决于数据的来源）。添加 Visualizer 也是个好主意，以便我们可以查看数据的变化。

13) 添加目标数据集

在菜单条中选择 Destination Data > Add Dataset，参数设置如下：

Destination Format ESRI Geodatabase (File-based)

Destination Dataset C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

Advanced Writer Settings

Writer Mode: UPDATE

Transaction Type: Edit Session



14) 添加目标要素类

在添加一个目标数据集时，FME 通常会询问你是否需要一个新的要素类（左图）。

点击‘No’，因为我们会使用源数据来产生目标数据。

15)复制目标数据

因为我们将 Parcels 数据集作为源数据集，我们可以利用它来产生目标要素类。右键点击源数据集，在弹出的菜单栏中选择“Duplicate as Destination”，打开 Feature Type Properties 对话框，检查 Parameters 选项卡设置是否如下：

Destination Settings

Table exists: Yes

Truncate Table First: No

DropTable First: No

Update Key Fields: OBJECTID

16) 连接要素类

连接 AttributeCreators 到 Parcels 要素类。

17) 保存和运行工作空间

保存和运行工作空间。

在 FME Viewer 或 ArcCatalog 中检查结果，你就能从 Visualizer 中查看到什么要素被添加或者删除。注意被添加的要素不含有所需的所有属性。

函数更新

FME 含有许多 Workbench 函数，专门用于数据库。它们显示在 Transformer 面板的 Database category 下面。这类 Transformer 一般用于查询数据库，但也能用来发送更新、插入或是执行属于 Geodatabase 的数据的空间关系。

对于这些 Transformer，每次输入要素，一个 SQL 或是其它数据库指令会被触发。如果通过一个查询从数据库中查询，输出要素可能是全新的。

这些 Transformer 有时更适合使用 writer 来执行更新，因为你可能希望只应用在一个对小的数据集的更改，或者你想使用一个特别的 where 从句（这个从句当你选择 writer 的 UPDATE 模式时是不能获取的）。

尽管如此，除非因为特殊的原因需要这些功能，你应该使用 writer 来代替。

SQLExecutor

SQLExecutor 是解决对数据库的 SQL 命令的主要的 Transformer。这些通常都是简单的选择指令，但是你也能够发布 insert, update 和 delete 指令。

这个函数是格式中立的，也就是说，你能使用任何数据库的格式来发布指令。

使用 FME 函数@Value()就能访问属性，如下：

```
delete from ADDRESS_POINTS
where PRIMARYINDEX = @Value(PRIMARYINDEX)
```

我们不推荐使用 Geodatabase 的 SQLExecutor，因为它主要在基础数据库中使用，而不是版本地理数据库，不能实际的处理你的数据。

ArcSDEQuerier

这个 ArcSDEQuerier 是一个发布 ArcSDE 数据库的查询指令的 Transformer。

最初这个 Transformer 是更新的主要工具，如更新数据库，所以支持模型设置（查询，更新 或 删除）。然而从对 SDE 的 writer 进行了一些改进后，它更适用于使用 writer 而不是 Transformer 来更新数据。不过在查询模式下的全套的空间交互是可获得的。



例 12 – 执行空间查询

在这个例子中，我们将提取 AddressPoint 数据的子集，把它输入到一个 DWG 文件进行字段检查。不过我们没有 TILE_NO，不能基于 CITY GRID 来提取它。为了达到这个目的，我们可以再次使用空间查询 AddressPoint 的信息，这一信息被保存在一个个人的 SDE 数据库中。

1) 创建一个工作空间

设置如下：

Source Format ESRI Geodatabase (File-based)

Dataset C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

Source Settings

Table Name: CITY_GRID

Where Clause: TILE_NAME='N26'

Destination Format Autodesk AutoCAD DWG/DXF

2) 放置一个 Transformer—— ArcSDEQuerier

按照向导的步骤使用下列设置：

Server: FMEUC2009

Instance: sde:sqlserver:FMEUC2009\sqlexpress

Dataset: CityData

User ID: FMEUC2009\Administrator

Password: a

Version: dbo.DEFAULT



左图：ArcSDE 的连接参数

Select the table: ADDRESS_POINTS

Mode: Query

Search Method: SDE_CONTAINS

Search Order: Optimize

3) 连接目标数据集

连接目标数据集到标志为 DBO.ADDRESSPOINTS 的端口。现在只有满足空间关系的要素才能经过这个端口输出。

4) 保存和运行工作空间

在浏览器中检查结果。与完整的数据表比较下，检查他们满足的空间关系。

5) 与 Clipper 比较

读取源数据表和通过经过一个 Clipper Transformer 的数据来得到所需要的数据，相同的处理被执行。用 Clipper 的方法来运行测试的工作空间，证明是非常慢的，这是因为所有的 ADDRESSPOINT 数据需要读入到工作空间，并传到 Clipper 中。作为数据库，使用 ArcSDEQuerier 可以更有效的做这一工作。

查看 C:\FMEData\Resources\UC2009_ESRI\Workspaces\example12_clipper.fmw，作为另一种方法的例子。



数据分发

进行数据分发的好处

为了能分发数据，通常需要能够从数据库中导出数据。这样就能简单的传送数据给那些不能以他们选择的方式直接访问数据库的人（如合同工），或者是简单的创建一个入口网站，从里面下载或使用流数据。

通过创建一个数据分发进程（这个进程是预定义的，但是可变通的）和提供给终端用户不同的选择，有下列好处：

-- 节约时间和资源：当用户能够为自己提供服务时，

只需要很少的资源就能处理数据传输需求。

-- 使用户更强大: 用户可以使用选项，在用户需要时选择他们以想要的格式来下载数据。

小提示:你可以在 Safe 网站上阅读一个典型的例子来学习数据的分发，网址是: http://www.safe.com/technology/documents/casestudies/CaseStudy_IHS.pdf。

用另一种格式导出数据，通常使用 FME Desktop 来执行。数据下载入口和数据流都需要安装 FME Server

使用 FME Desktop 进行数据分发

使用 FME Desktop 进行数据分发意味着或者你运行一个工作空间来导出数据，或是你作为 FME 管理员创建一个工作空间让其他人使用。

简单的输出数据

对于简单的数据，你所要做的是创建一个工作空间来从要求的数据库表中读取数据，然后将它写入到要求的格式中。

例 13 – 写入数据到 AutoCAD

这个例子首先进行的是一系列数据分发技术的检查。

这里我们将一个 Geodatabase 要素类转化成 DWG 格式，不过我们只需要 City Grid 表中的 TILE_NAME 为 N26 要素的图层信息。

1) 创建工作空间

创建一个工作空间，来读取一组 Geodatabase 要素类，并输出到另一种格式，设置如下：

Source Format ESRI Geodatabase (File-Based)
Source Filename C:\FMEData\Resources\UC2009_ESRI\CityData.gdb
Source Settings
Table List: City_Grid
Where Clause: TILE_NAME='N26'
Destination Format Autocad DWG/DXF

2) 添加第二个数据集

在菜单工具条中选择 Source Data > Add Source Dataset，在工作空间中添加另一个 Geodatabase 的 reader。因为 WHERE 从句不能应用在 Water 信息上，我们需要两个 reader，又因为 reader 会读取每个要素来查找 TILE_NAME 属性，会减慢读入的速度。这一步的设置如下：

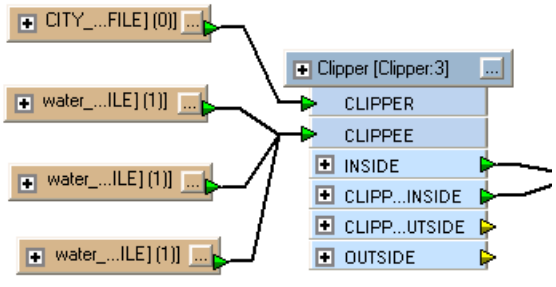
Source Format ESRI Geodatabase (File-Based)
Source Filename C:\FMEData\Resources\UC2009_ESRI\CityData.gdb
Source Settings
Table List: Water_distribution_lines
Water_distribution_nodes_inserts
Water_meters

3) 放置一个 Transformer—— Clipper

在工作空间中放置一个 Clipper 转换器。连接的路径是从 City_Grid_Poly 源数据集输出的要素连接到这个 Clipper 的 CLIPPER 输出端口，从 Water 图层输出的要素连接到 CLIPEE 端口。

现在看起来应该像这样（右图所示）

我们想收集从 INSIDE 和 CLIPPED INSIDE 端口输出的数据。



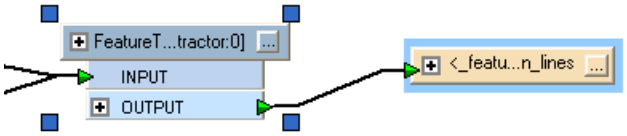
4) 放置一个 Transformer –

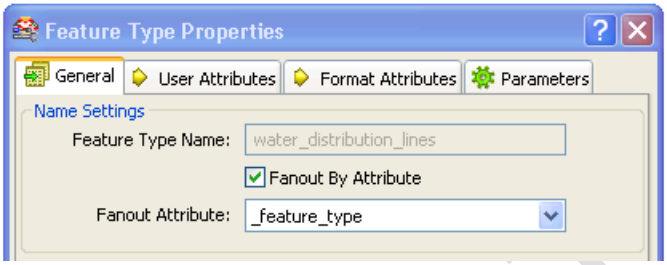
FeatureTypeExtractor

我们想输出信息到一个 dwg 文件，这个文件中的每个图层代表 Geodatabase 的一个图层。为了这个目的，我们会列出基于属性的目标数据，这些属性含有将要读入的要素类的名称。这里，我们添加一个 FeatureTypeExtractor 来提取要素类名称——默认值是 feature_type.。

5) 列出目标数据的属性

连接目标数据库。打开 Feature Type Properties 对话框，设置 Fanout Attribute 的值为 attribute _feature _type。





6) 保存和运行工作空间

在 FME Viewer 中检查结果。它会创建一个含有许多图层的 Autocad DWG 文件——每个图层都表示了 Geodatabase 的一个图层，数据应该只表示了一个城市。

7) 产生 Autocad 块

我们的原文件在节点上有 autocad 块。当我们加载数据到空间数据库时，会产生一个新的图层，称为 water_distribution_nodes_inserts，它保留了插入点的 X、Y 的位置。我们使用这一图层来创建输出的块。因此需要过滤这个图层的数据，然后将它写入到一个新的要素类中。右击点击目标数据集，然后在下菜单中点击 Duplicate 选项。设置如下的属性：

Feature Class Name: water_nodes

Fanout By Attribute: unset

Expose attributes: autocad_block_name

autocad_entity

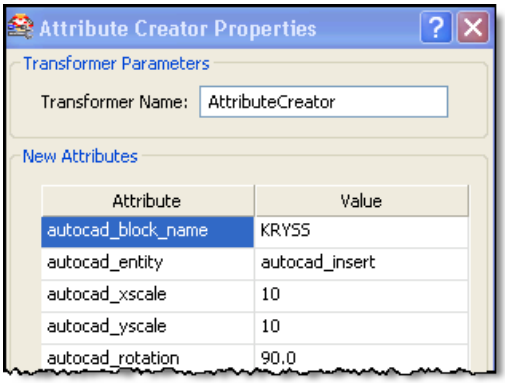
autocad_xscale

autocad_yscale

autocad_rotation

8) 放置一个 Transformer – AttributeCreator

添加一个 AttributeCreator 来创建上面暴露的属性的值。



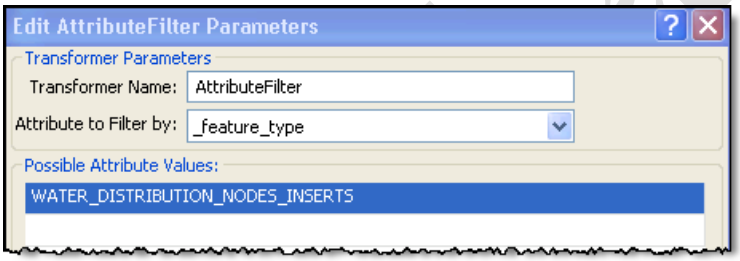
9) 增加一个模板到目标数据集的 Writer

对于 FME，为了找到匹配上面属性的块，我们需要在 writer 中指定一个模板文件。在导航面板中选择任何一个原始的水系分布文件作为模板。

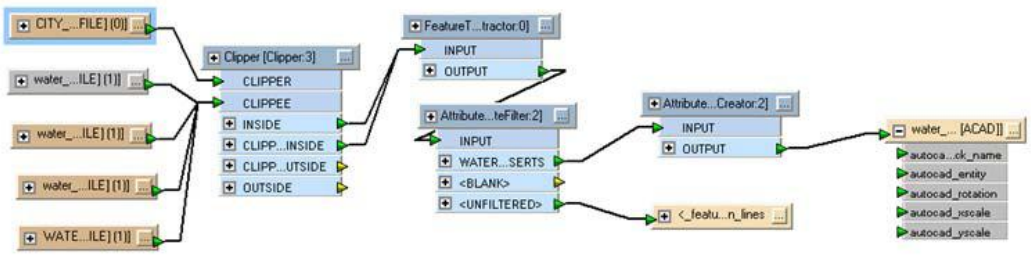


10) 放置一个 Transformer – AttributeFilter

为了过滤 water_distribution_nodes_inserts 要素，我们需要一个 AttributeFilter 转换器，使用一个要素类型的定义来过滤。将它放到 FeatureTypeExtractor 和 AttributeCreator 之间。



现在你的工作空间就是这样的（如下图）



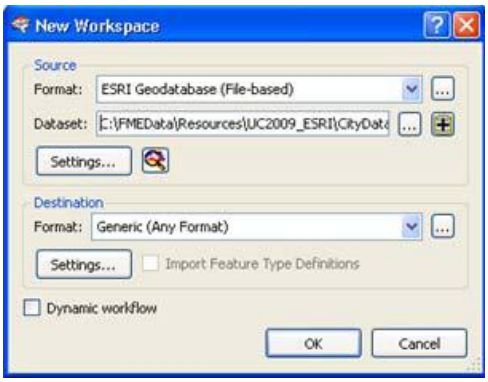
11) 保存和运行 workspace

在 FME Viewer 中检查结果。它会创建一个含有许多图层的 Autocad DWG 文件，每个图层上对于每个节点位置都有一个块。

针对其他用户的工作空间

当你为其他用户创建工作空间时，可能用户并不是 FME 专家，你就需要添加不同的选

项，这样他们在不必编辑工作空间的情况下，也能调整数据输出。

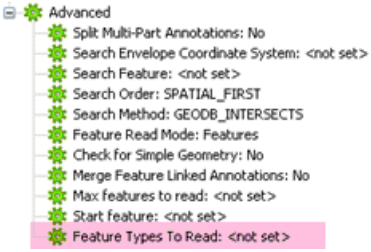


Generic Writer 对于自定义输出格式，它是一个很好的工具。它是这样一种 writer，直到运行时，它才会定义格式。通过发布参数，你就能设置你希望的格式，按照提示，用户能选择他们需要的输出格式。

上图：使用 Generic 格式的 writer 设置的工作空间。

Feature Types to Read 参数

Feature Types to Read 是一个大多数 Reader 都可以获取的参数。它允许用户选择要从源数据集中读取哪些 Feature Types（也就是表）。



左图 ：Feature Types to Read 是一个参数，你可以从导航窗口中的一个 reader 的 Advanced 部分中找到它。

例 14

1) 创建工作空间

使用新的工作空间对话框创建一个工作空间来读取一组地理数据的要素类，并输出到 Generic writer 中，这样在这个工作空间中用户就可以选择他们感兴趣的格式了，设置如下：

Source Format ESRI Geodatabase (File-Based)

Source Filename C:\FMEData\Resources\UC2009_ESRI\CityData.gdb

Source Settings

Table List: CITY_GRID, CITY_PARKS, PARCELS

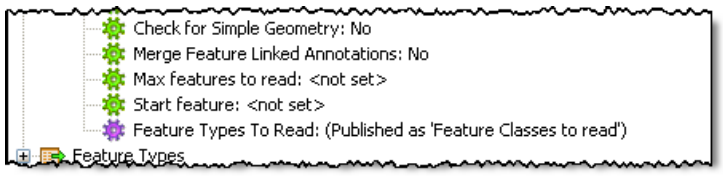
Destination Format Generic (Any Format)

2) 设置 Feature Types to Read 参数

在 Workbench 的导航窗口中找到 Feature Types to Read。

右键点击这个参数，选择 Publish Parameter 选项。

更改提示“Feature Types to Read”为“Select Feature Classes to Read”，让非 FME 用户清楚的知道他们的要求。



3) 删除 Published Parameter

只要清空提示框，删除与源地理数据库文件相关的 Published Parameter 就行。

4)运行工作空间

使用 Prompt and Run 选项来运行工作空间——或者先保存，然后用 FME Universal Translator 来运行。你会被提示选择读取哪些数据表，将它们写入到什么格式。尝试不同的组合看看输出的数据会有什么不同。

Dynamic Feature Types to Read



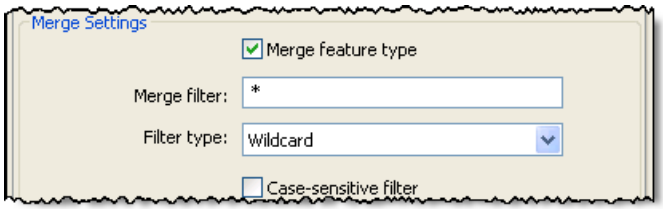
你会注意到，在 Feature Types to Read list（左图）中出现的数据表都来自工作空间自身的。

在数据库中有许多其它的数据表(例如 NESTS 或 EGGS)，但是我们没有被提示来选择它们。

FME2010 弥补了这个缺陷，允许你选择数据表（即使这些表并没有在工作空间中定义），要使用新的 Feature Types to Read 的 Dynamic 版本。

5) 设置 Merge Features Filter 参数

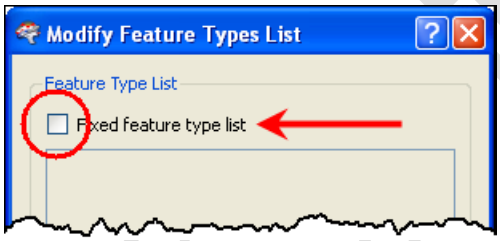
设置 Merge Feature Filter 非常有必要，这样就能通过 reader 读入要素类，不必考虑它们的结构。使用一个通配符，如‘*’。

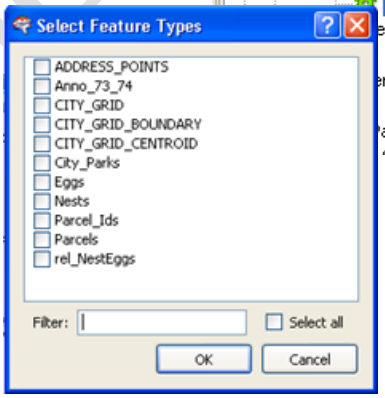


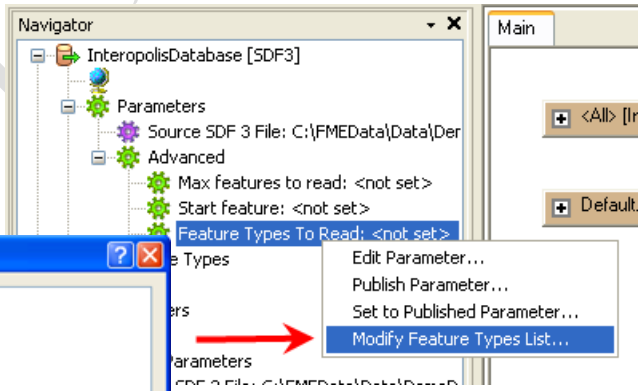
左图: 为了允许未定义的要素类能通过工作空间，需要选中 Merge Feature Type 选项。

右图:现在，Feature Types to Read 参数有一个新的选项：Modify Feature Types List。

下图:在 Modify Feature Types List 对话框中，确定没有勾选“Fixed feature type list”选项。







右图：现在 Feature Types to Read 选项会提示用户数据库的所有数据表。

转换的可能性（Transformation Possibilities）

和其它 FME 工作空间一样，数据库在读取工作空间时，能从其它源数据库中合并数据，进行数据转换。

重新符号化（Resymbolisation）

转换数据的一个主要原因就是为了提供一套合适的要素符号。空间数据库趋向于存储数据而忽略符号，除非被连接到一个特别的应用程序。例如，ArcSDE 会储存要素的颜色，但是 Oracle 或 PostGIS 数据库则不会。

例 15

这里，我们将继续操作数据分发的工作空间

1) 打开工作空间

打开例 14 中的工作空间

2) 添加 KMLStyler Transformer

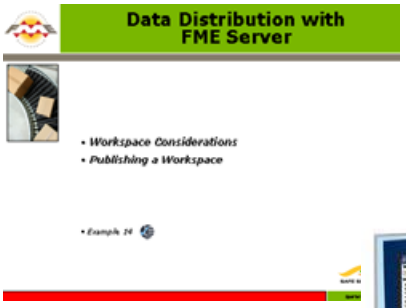
在工作空间中，对于每个要素流添加一个 KMLStyler 转换器。

使用这些 Transformer 就能用不同颜色（例如，红色代表公路，蓝色代表运输线，白色代表机场）和不同的线厚度来表示不同的要素。

3)运行工作空间

运行工作空间，在 Google Earth 中验证新的要素颜色。

使用 FME Server 进行数据分发



使用 FME Server 进行数据分发涉及了发布预定义的工作空间到一个服务器供别人使用——即使用户不了解或是没有 FMF。

使用 FME Server 进行数据分发，通常意味着你创建一个工作空间，让其他人使用。使用服务器的构架，从工作空间输出数据甚至能发送到另一个应用程序。



工作空间的考虑（Workspace Considerations）

在发布一个工作空间到 FEM Server 上时，你要考虑许多问题。

源数据的位置

首先，服务器要能够访问源数据。事实上使用工作空间访问源数据是非常简单的，因为你仅仅需要创建客户端和正确的连接。

目标数据位置

第二，为了能在服务器上运行工作空间，必须要能访问输出位置。这可能不是一个像我们正在使用的 C:\FMEData folder 格式的本地目录，因为服务器不能直接访问它。它必须是一个 UNC (Universal Naming Convention)路径，如\\mycomputer\myoutput。

当数据以流的方式访问或者使用一个数据下载服务时，访问一个输出位置不是一个必要因素。

在没有输出路径的情况下，FME Server 会提供一个临时的数据位置。

发布一个工作空间

在 Workbench 中打开工作空间，使用菜单 command File > Publish to Server，就能发布工作空间到 FME Server。

你需要一个要发布到的能获取连接的服务器名称和端口号（通常端口号是 7071）。一旦进行了连接，你就可以选择将工作空间放到一个模板库（模板库是一个工作空间、脚本和数

据的目录）中。

最后，你需要指定你想要注册你的工作空间到哪个网络服务。例如，一个数据下载服务能够提供下载压缩文件， KML Network Link 服务提供像 KML 数据一样的连接，这些数据会定期被更新。

例 16

现在我们发布我们分发工作空间到 FME Server。

1) 打开和发布工作空间

打开例 14 中的工作空间。

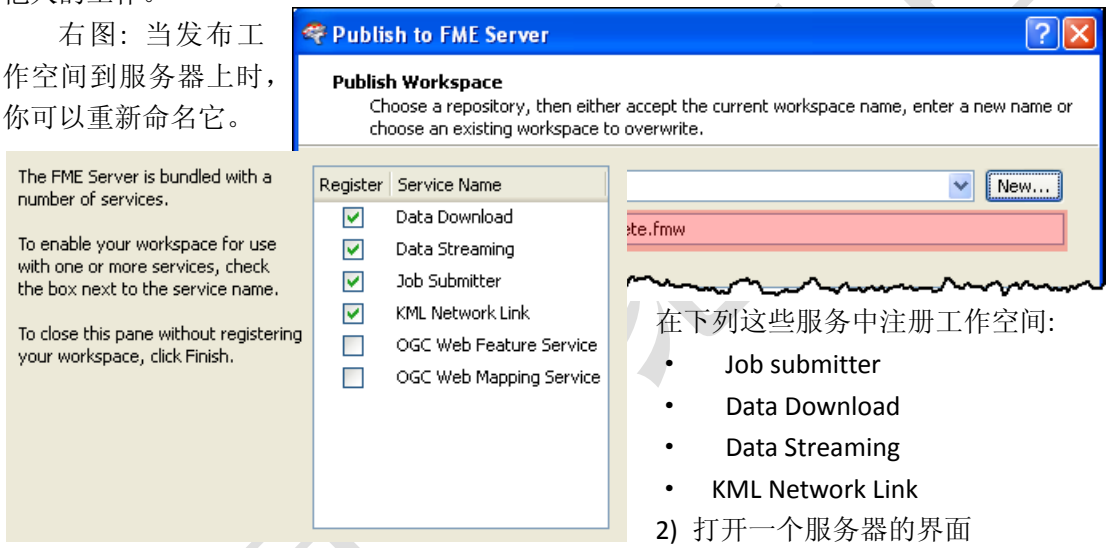
选择 File > Publish to Server，发布工作空间到指导老师的 FME Server 上。在 Safe Software 培训室中，这个连接设置如下：

Hostname: trm17

Port Number: 7071

发布工作空间到一个名为 Training 的模板库中。一定要对工作空间重命名，避免覆盖其他人的工作。

右图：当发布工作空间到服务器上时，你可以重新命名它。



在下列这些服务中注册工作空间:

- Job submitter
- Data Download
- Data Streaming
- KML Network Link

2) 打开一个服务器的界面

访问 http:\\trm17\fmeserver，在网站浏览器中打开服务器的界面。在 Data Downloads 服务中定位发布的工作空间。运行它以证明它被正确的创建。

NB:在做这些之前，确定指导老师已经把源数据加载到数据库是非常重要的！否则处理完成会发送一个成功转换的报告，但是没有输出!

3) 作为一个 Job Submitter 服务运行

JobSubmitter 服务只是运行工作空间，并将数据输出到定义的位置。

它作为一个高级的操作，在这个服务下运行工作空间。你需要哪些安装来确保数据会被发送回你的电脑呢？

课程回顾（Session Review）



这个单元被设计来帮助你使用 FME 在 ESRI 的 Geodatabase 中进行操作。

从这个单元中你能学到什么？

下面这些知识点就是你能学到的：

理论知识

- FME 处理 ESRI Geodatabase 中的所有指定要素类
- 和 FME 的其它功能一样，Geodatabase 使用层次结构（hierarchical structure）来控制转换（包括 R/W parameters, Feature Type parameters, Format Attributes）
- 使用数据库本身可以执行哪些操作（例如剪切源数据），比起读取整个数据表或是在 FME 中处理更高效。

FME 技巧

- 使用 FME 把数据写入到 Geodatabase
- 使用 FME 检查 Geodatabase 中的数据
- 操作特别的要素类，例如，注记（Annotation）、 关系（Relationship）、 和几何网络 （Geometric Networks）
- 进行表级别（Table-Level）和要素级别（Feature-Level）的更新
- 使用数据库相关的转换器（Database-Related Transformers）在 Geodatabase 中处理数据
- 在空间数据库中进行变化检测
- 使用 FME Desktop 或 FME Server 来分发 Geodatabase 中的数据

附录 A – 兼容性和安装问题

FME 2009 为了解决版本兼容的问题，FME2009 中，FME 不需要安装要求的库来读写 ArcSDE。你应该代替获取库，而不是从 ESRI 产品中获取，来保证 FME 和 SDE 数据库间的兼容性。有许多方法可以获取需要的库：

1）安装 ArcGIS Desktop 9.3；或

2) 安装 SDE C SDK（在你自己的 ArcGIS Server 9.3 DVD 上可获得），设置下面描述的环境变量；或

3）安装 ArcEngine，检查下面描述的环境变量；或

4）安装 ArcGIS Explorer，检查下面描述的环境变量。

如果没有设置系统环境变量的话，你需要如下的设置:

Variable Name - ARCGISHOME

Value - <ArcGIS Directory>\arcsde\或

Value - <ArcGIS Directory>\arcgis\

(也就说，指出哪些文件夹包含 SDE 的.dll 文件)

FME 2008 是用旧版 ArcGIS 9.2 库进行编译的，因此不能与 ArcGIS Desktop 9.2 SP5 兼容。通过以下的操作，就能避免这样的问题：

-- 在 FME 安装目录下移除（或重命名）下面三个 dll 文件:

- sde.dll
- pe.dll
- sg.dll

--为了在 AcrGIS 安装中正确定位这些 dll 文件，我们有必要添加 ArcGIS 的 bin 目录到你的 PATH 变量中。

- 如果你已经有了一个 ARCGISHOME 变量，就使用%ARCGISHOME%\bin。
- 如果你已经有了指向 ArcGIS/ArcSDE/ArcEngine 安装的其他的变量，使用和添加这个 bin 文件目录。

或者添加你的包含\bin 目录的 ArcGIS 安装位置。如果你同时安装了 FME2008 和 FME2009，那么你要保证做了上面两个步骤——也就是，你需要添加到 PATH 变量和获取已有的 ARCGISHOME 变量——但是要注意在目录树中的指向会稍微有些差别。

查看我们的 fmepedia 网站获取更多的信息，网址为：http://www.fmepedia.com

附录 B – 有用的调试资源

当运行一个有问题的 Geodatabase 时，FME 报告 ArcObjects 错误。这通常包含了 ArcObjects 的错误号和错误信息。我们会发现一般的错误信息并不能正确的解释错误号码，下面的网址对解释这些错误号是有用的：

http://resources.esri.com/help/9.3/ArcGISEngine/ArcObjects/esriGeoDatabase/fdoError.htm

在我们的 fmepedia 网页上，有一些其它的错误信息的解释：

http://www.fmepedia.com/index.php/Troubleshooting_Geodatabase_FeatureClass_Creation

FME Readers and Writers 手册中也有大量的关于 Geodatabase 的要素表达的信息和所有格式属性定义的信息。

附录 C – 性能考虑

在使用 Geodatabase 进行工作时，查看 FMEPEDIA 网页的下列文章可以获取更多详细的建议和性能的改进。

http://www.fmepedia.com/index.php/How_can_I_improve_the_performance_of_FME_when_readi ng/writing_to_Geodatabase

-- 更改 Features to Write per Transaction 设置来延长提交处理事务的间隔。

-- 如果可能，把 Transaction Type 设置为 TRANSACTIONS 或 NONE ，代替 EDIT_SESSION (MDB, GDB) 或 VERSIONING (Enterprise Geodb)

-- 写入现有要素类而不是用 FME 来创建表

-- 将 Simplify Geometry 设置成 NO，会加快几何的写入。

-- 通过一个 reader 读取数据时，如果你不需要几何网络相关的网络信息，就设置 Ignore Network Info 为 YES。

-- 使用 Search Envelope 就能在读取数据时，减少工作空间的数据流量

-- 确定任何一个 Where 从句都能应用在所有将要被 reader 读入的要素类上

-- 如果你不需要关系要素类，将 Ignore Relationship Info 设置成 YES。

-- 避免或最小化注记要素的数目——如果注记要素也参与了关系类时，写入要素就会非常慢

-- 只设置到你要求的精度。如果你需要的是 32 bit 的精度，就将 writer feature type 参数设置成 32 bit，而不是 64 bit。

-- 对于企业级地理数据库（Enterprise Geodatabase）和 ArcSDE，你可以通过删除空间索引和把层（空间列）放入 LOAD_ONLY 模式来减少加载时间。

-- 在 FME 工作空间的导航树中，保证你的数据库的 writer 是第一个 writer(右击 writer, 使用 Move Up 按钮)。为了第二个和随后的 writers，FME 会缓存数据，这样首先保证数据库的 writer 能直接传送要素。

-- 使用 ArcSDE 的 reader/writer。AcrSDE 的 writer 允许缓冲的写入，这样就能干净的加载数据，改进数据加载性能。在 SDE 中，你也能设置 load_only_mode（关闭索引），然后在转换完成后重新设置为正常模式。