目录

引言 2
培训数据 2
在 CAD 文件中存储属性 3
在一个外部的表中存储属性 6
存储属性为注记 10
创建面标注（Create Area Labels） 10
从面标注中提取（Extract from Area Labels） 19
创建线性标注 28
从线性标注中提取 30
用符号分类属性 33
创建符号 34
从符号中提取属性 38
结束 39

北京世纪安图

引言

本课程涵盖了将GIS数据转化为CAD数据或CAD转化为GIS的方法,并强调了传统的CAD绘图上，为属性存储的所作的努力。我们将探讨如何将GIS属性存储为CAD属性,连接到外部表、注记及符号体系和如何将提取的数据在不丢失信息的情况下写回地理信息系统(GIS)中。

专题讨论会的前提条件

参加者都应该对FME Desktop、创作工作空间和更常见的FME转换器比较了解。

培训数据

我们将使用以下FME培训的数据集:

- Data\Properties*.mif
- Data\Roads\MajorRoads.dgn
- Data\Roads\RoadLine.mif

北京世纪安图

在 CAD 文件中存储属性

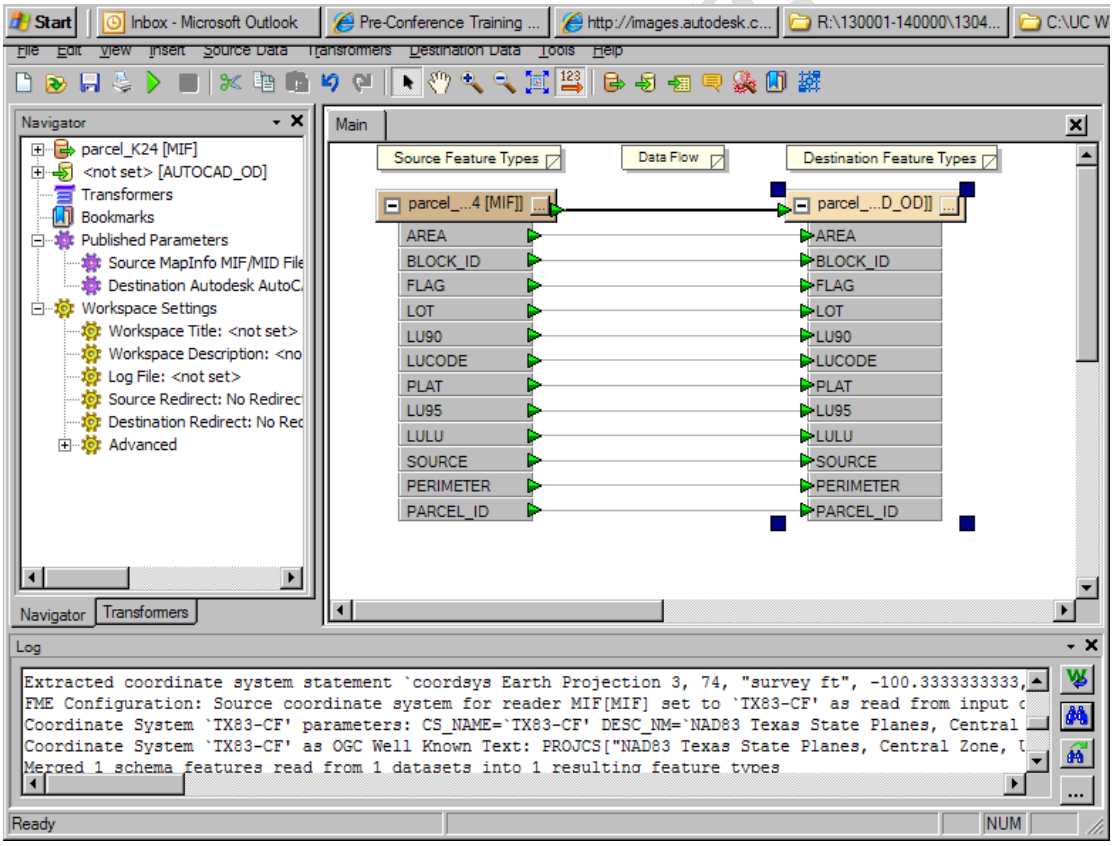
AutoCAD Map3d 和较新版本的Bentley MicroStation中，都允许在要素自身中存储属性。用这种方法来存储GIS属性是非常简单的。

1、 启动Workbench，打开工作空间对话框来创建一个新的工作空间。

2、选择MapInfo MIF/MID作为源格式，选择Data\Properties\parcel_K24.mif作为源数据集。

2、 选择Autodesk AutoCAD Map 3D Object Data作为目标格式，点击OK来创建工作空间。

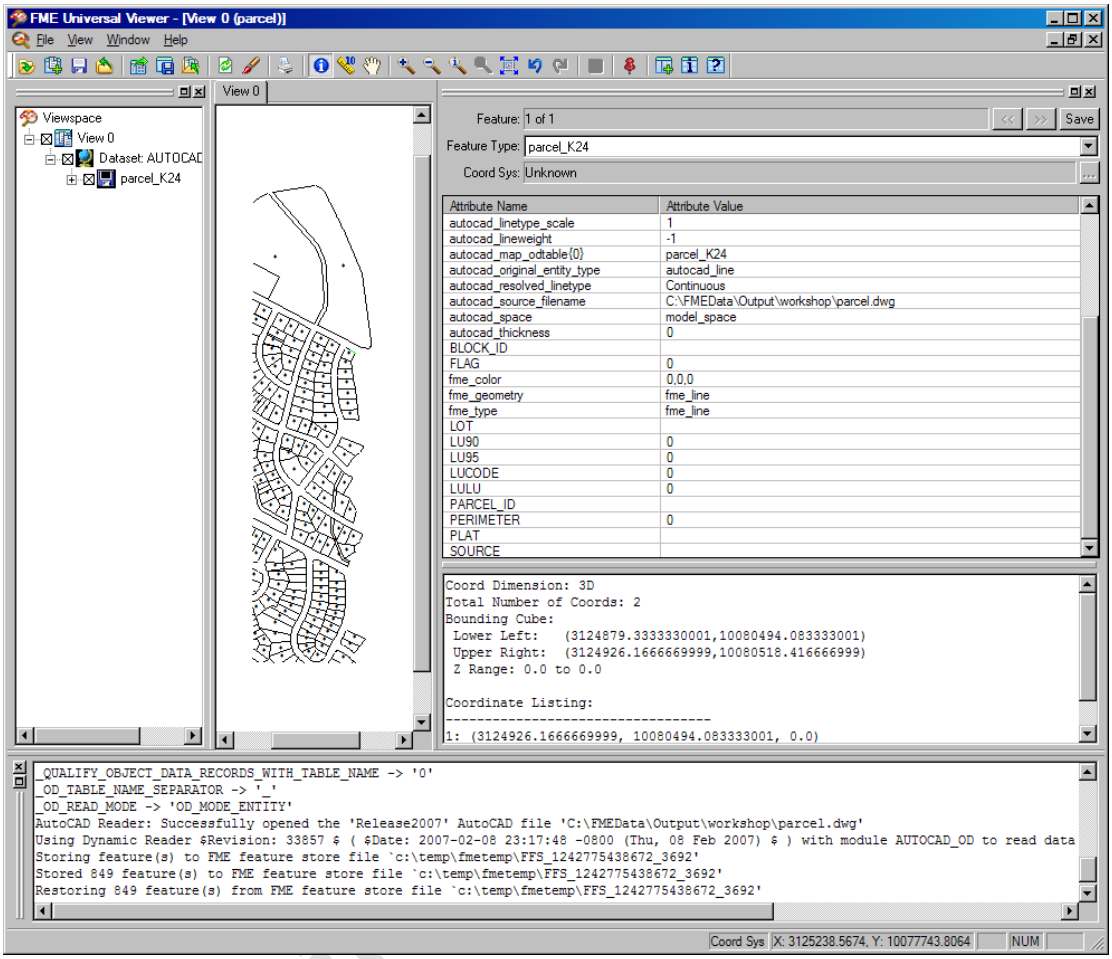
你的工作空间应该看起来像这个样子：



4、 点击绿色播放键来进行转换

5、设置目标文件为：Output\workshops\CAD\parcels_object.dwg。

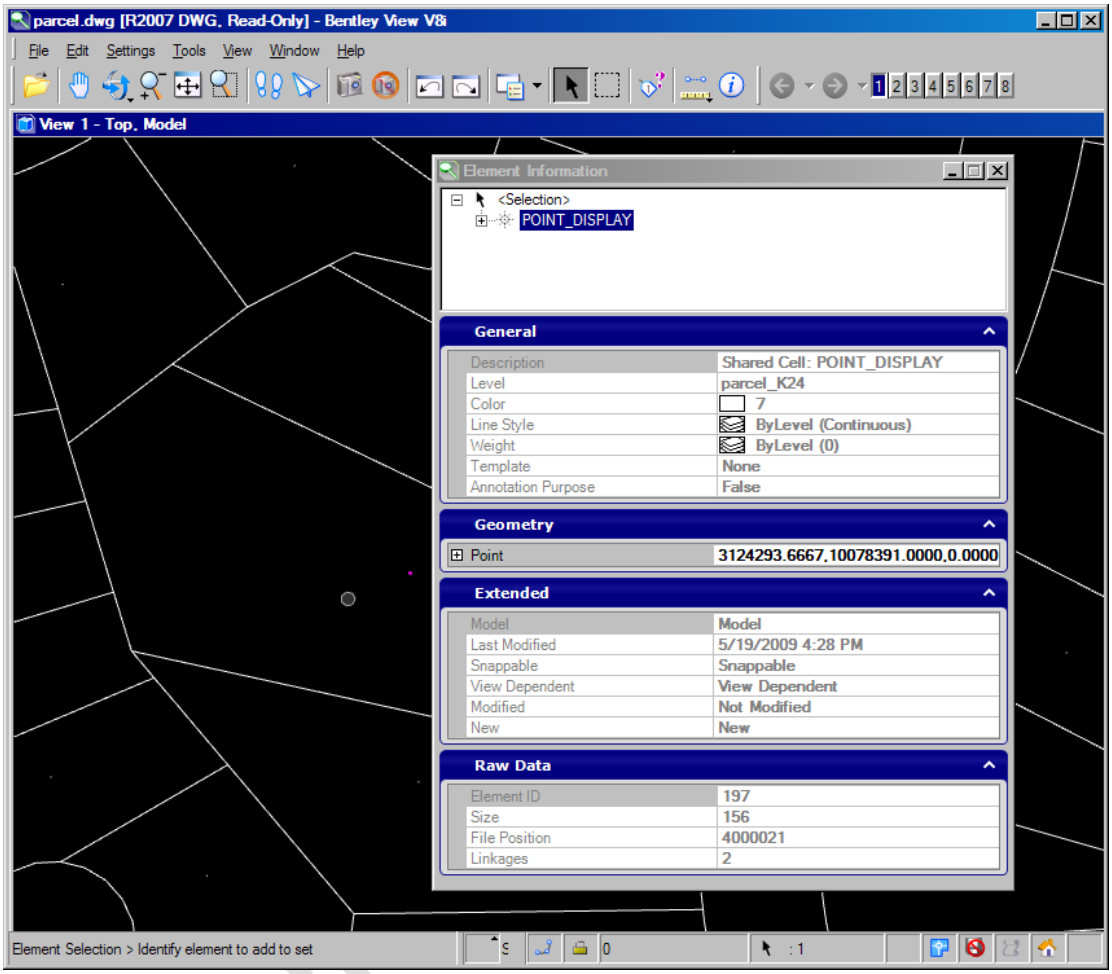
6、一旦转换完成，在FEM Viewer中设置文件格式为Autodesk AutoCAD Map 3D Object Data，来打开源文件，如下所示：



你会注意到，所有的属性都赋给了AutoCAD数据。

7、关闭FME Viewer，打开Bentley View。在文件打开对话框中，选择parcels_object.dwg

文件。点击菜单上的①按钮来打开属性对话框，点击一个质心来查看它的属性。



注意，地块的属性不能被看到。这个对象数据只有在AutoCAD Map3d或FME Viewer中才

是可见的，这就限制了它的应用。

AutoCAD也有一个选项来存储数据作为扩展实体数据（Extended Entity Data）。这些属性

可以在正式的AutoCAD中读出，虽然AutoCAD没有提供一个接口来做这些。Safe公司也提供了

一个免费的VBA宏允许你添加或编辑扩展实体数据，这个宏可以在下面网址上下载：

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

在一个外部的表中存储属性

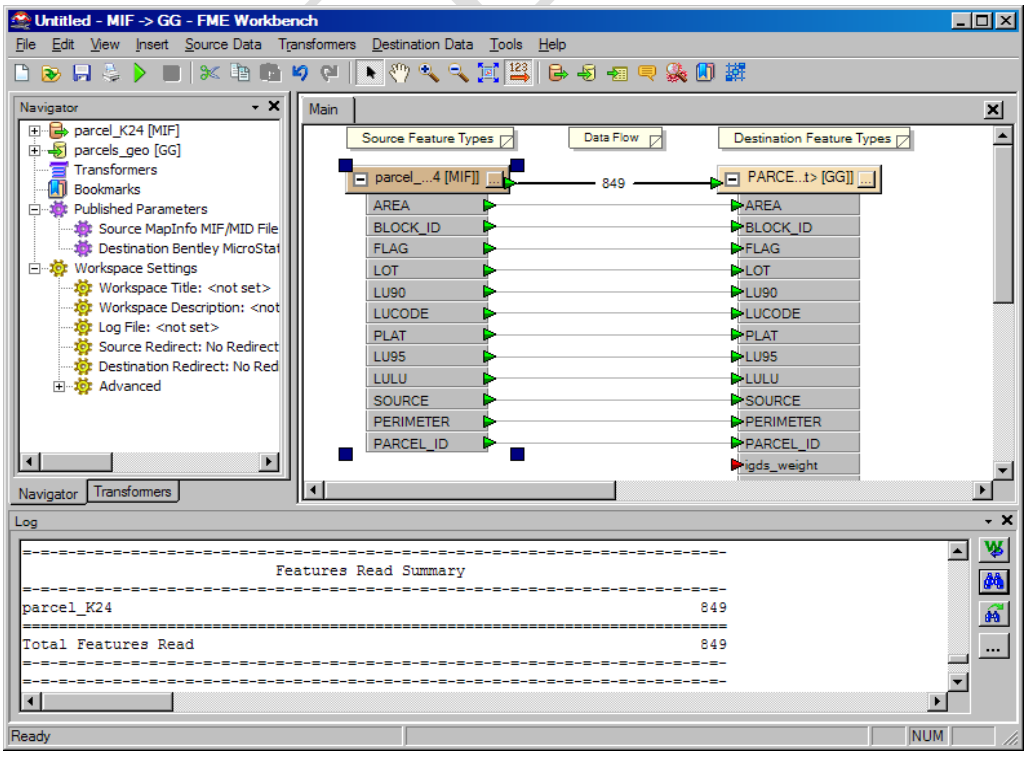
MicroStation也允许在一个外部的表中存储属性，通过igds_linkage格式属性连接到几何要素上。同时，手工的创建一个连接是可能的，最简单的方法是使用FME的Bentley Geographics读写器。通过Geographics writer创建的连接和表可以实际应用在MicroStation中，甚至在没有附加地理信息的情况下。

1、 启动Workbench，打开工作空间对话框来创建一个新的工作空间。

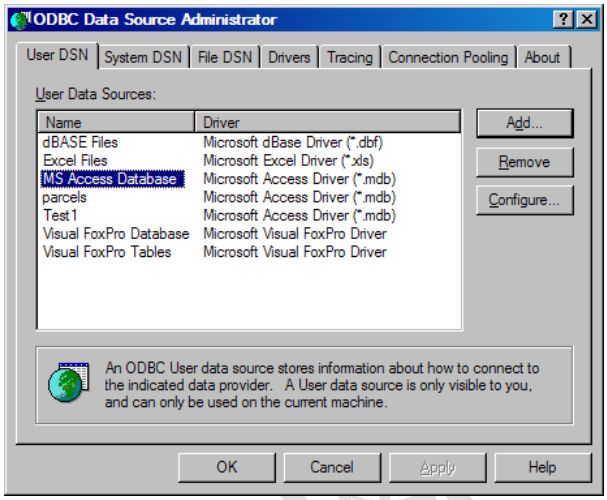
2、 选择MapInfo MIF/MID作为源格式，选择Data\Properties\parcel_K24.mif作为源数据集。

3、选择Bentley MicroStation GeoGraphics作为目标格式，点击Settings按钮。设置Database Type为MDB，点击Access MDB File选项后面的…按钮。设置输出目标数据的位置为Output\workshops\CAD\parcels_geo.mdb。保留所有的其它设置为默认值，点击OK，退出这个对话框。点击OK关闭New Workspace对话框，并创建了一个工作空间。

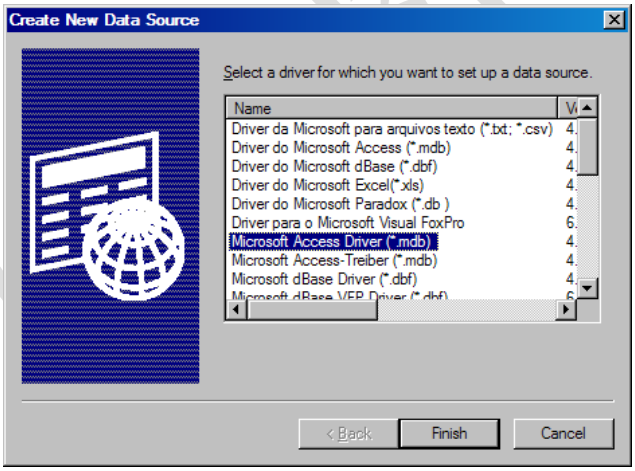
4、点击运行按钮，开始转换。当FME为这个MicroStation文件请求一个目标文件时，设置这个文件为Output\workshops\CAD\parcels_geo.dgn。



在MicroStation中查看属性



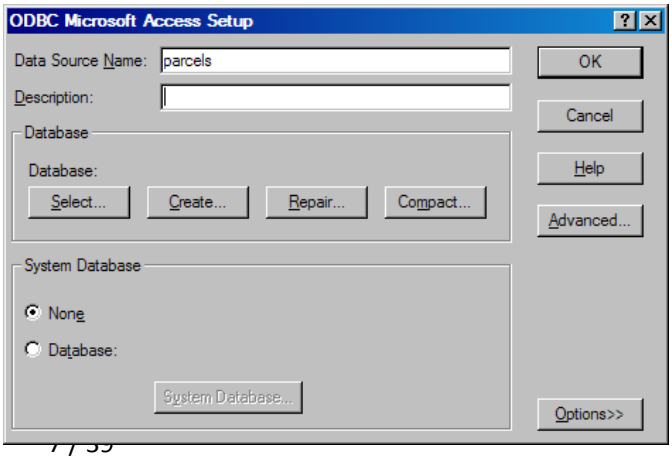
5、创建一个新的ODBC 连接。打开控制板 - 管理工具 - 数据源(ODBC).



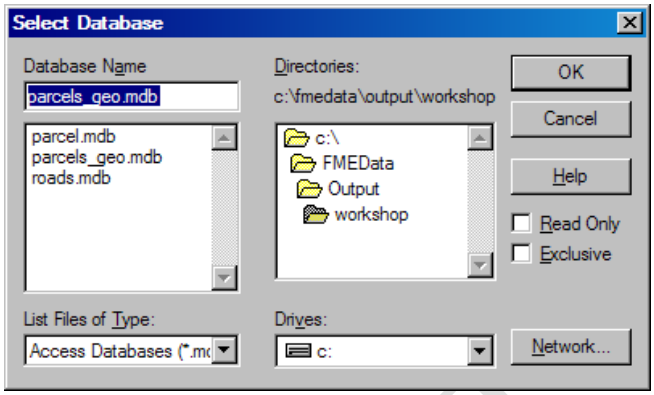
6、 选择User DSN选项卡，点击MS Access Database ，然后点击Add… 按钮。

7、选择Microsoft Access Driver(*.mdb)点击Finish按钮。

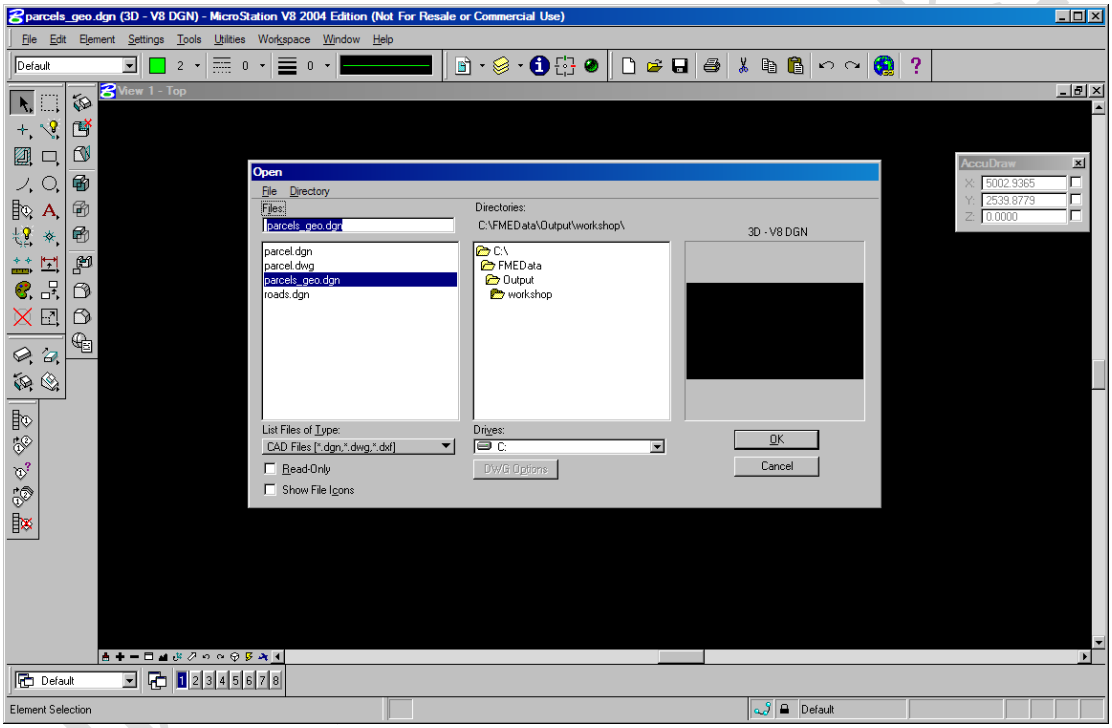
8、 设置Data Source Name 为 parcels，选择 Database中的Select…按钮。



9、 选择工作空间创建 parcels_geo mdb 文件，点击三次 OK 退出 ODBC Data Source Administrator 界面。



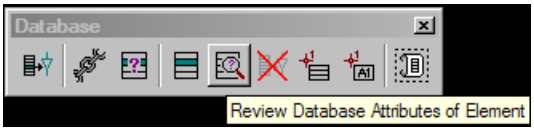
10、 启动MicroStation ，然后打开工作空间创建的parcels_geo.dgn文件。



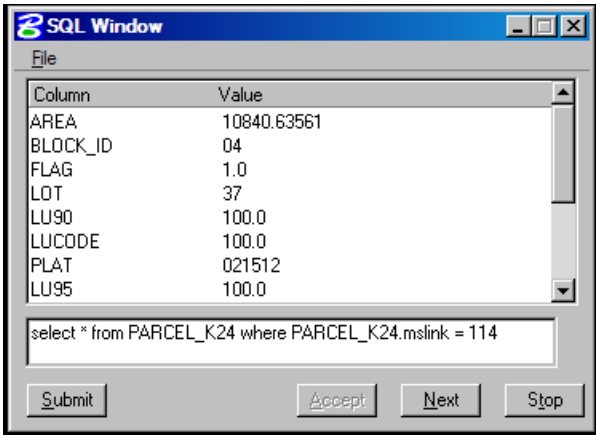
11 从菜单项中，选择 Settings – Database – Connect 菜单。选择 ODBC 作为数据库服务（Database Server），输入 parcels 作为连接字串（Connect String）

12 在菜单中，选择Tools – Database来打开数据库菜单条。

13 在Database菜单条上选择Review Database Attributes of Element图标。



14、选择地块的质心来显示它的属性。



北京世纪安图

存储属性为注记

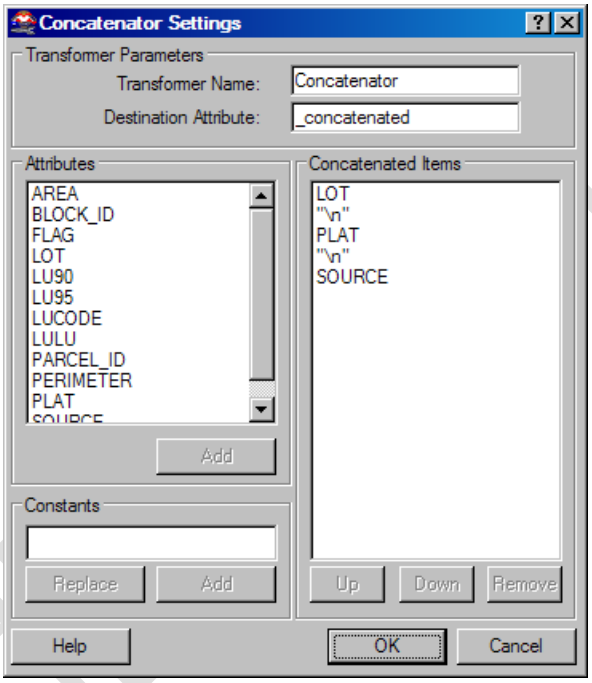
很多时候，数据从 GIS 转换到 CAD 来创建制图输出。在这种情况下，GIS 属性被转化为 CAD 文件中的注记。并且用来制图的 CAD 文件也经常作为 GIS 数据的源文件。在接下来的几个工作空间中，我们将探讨创建和提取多边形和线要素的属性作为注记的例子。

创建面标（Create Area Labels）

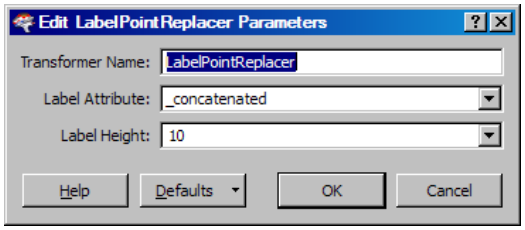
在这个练习中，我们将使用三个属性在多边形中创建标注。

1、启动Workbench，使用MapInfo MIF/MID作为源格式，Autodesk AutoCAD DWG/DXF作为目标格式创建一个工作空间。设置源数据为：Data\Properties\parcel_K24.mif。

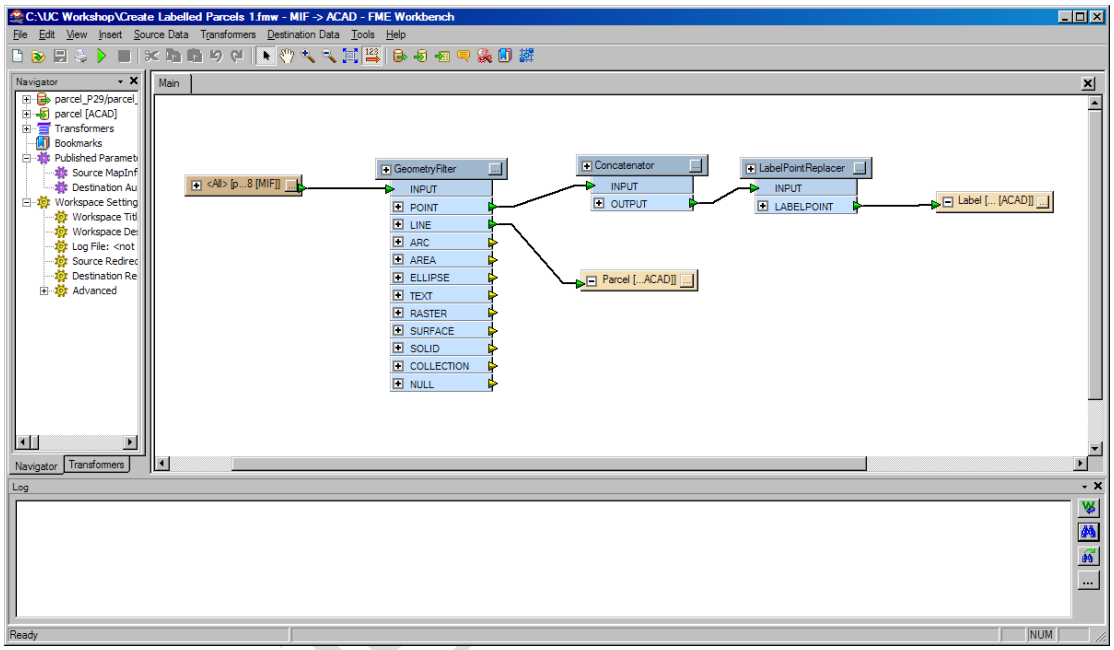
2、添加一个几何过滤器（GeometryFilter）来分离出线和点。把点连接到Concatentor转换器上，并添加属性LOT, PLAT 和SOURCE到这个连接，通过“\n” 添加一个回车换行来分开。



3、添加一个LabelPointReplacer转换器来连接Concatenator的输出。设置Label Attribute为_concatenated，Label Height为10。

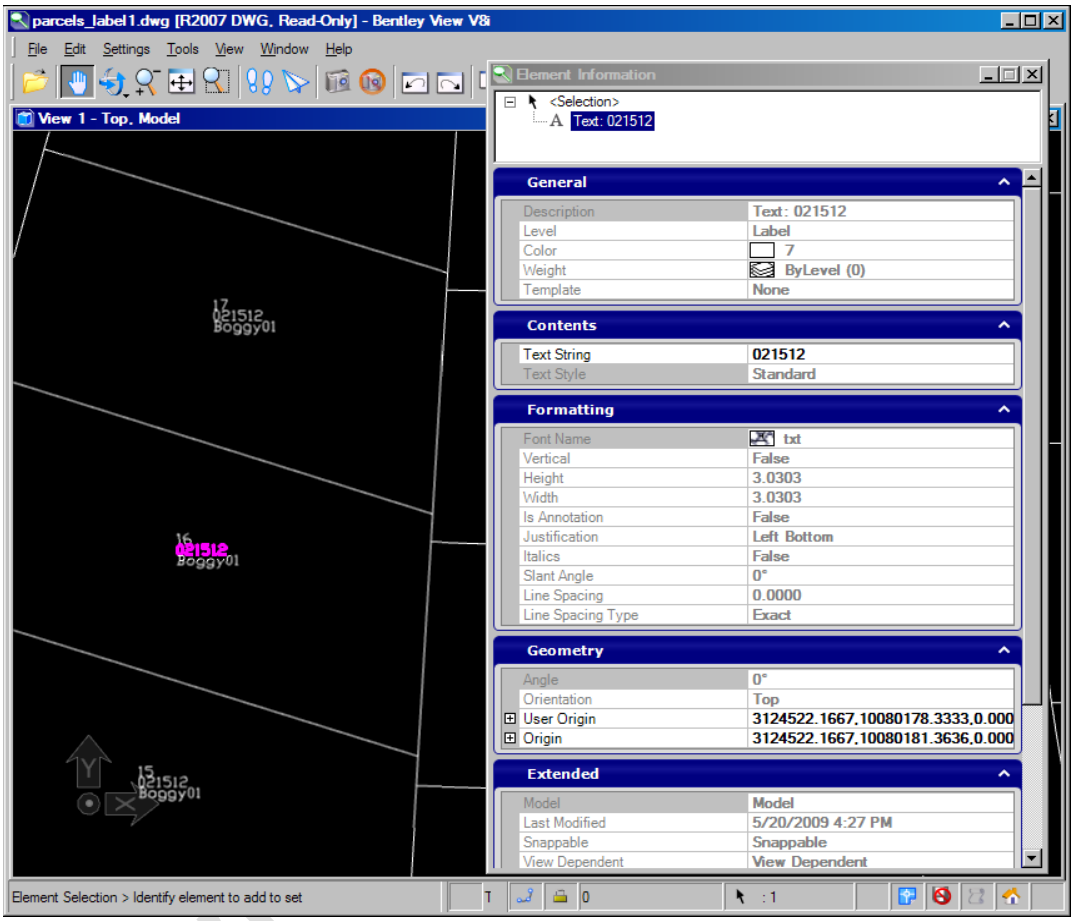


3、 创建两个目标要素，Parcels 和Labels。连接Parcels到 GeometryFilter的Line输出端口，Label到LabelPointReplacer的Labelpoint输出端口。



5、运行这个工作空间，设置目标数据集为：Output\workshops\CAD\parcels_label1.dwg。

6、在Bentley View中打开文件parcels_label1.dwg，验证标注文本。



7、首先我们注意到我们已经设置文本高度为10，而文本的实际大小为3.03。FME通过回车把这个文本打碎成多个部分，但是对于标注把文本大小的设置作为整个文本的大小来对待。

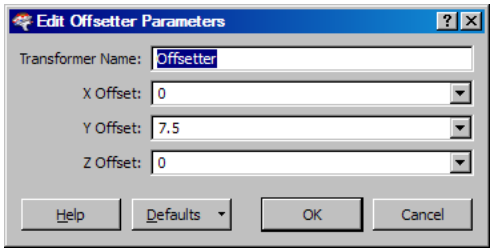
8、为了对文本大小有更大的控制，我们对工作空间进行一些修改。移除Concatenator，复制两个LabelPointReplacer，垂直排列。

修改上面的LabelPointReplacer转换器的Label Attribute值为LOT，中间的值为PLAT，下面的值为SOURCE。更改所有的Label Height为5。

9、我们可以将这些新的转换器连接起来，运行这个工作空间，不过所有的标注将会放置在同一点上。所以我们将给这些标注一个位移，使它们易于阅读。

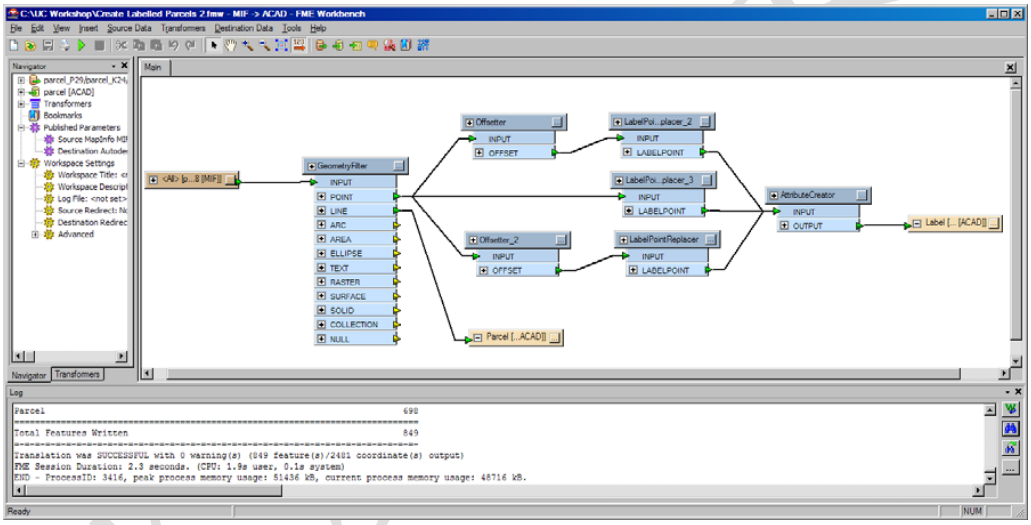
在 GeometryFilter 和顶部的 LabelPointReplacer 之间添加一个 Offsetter 转换器。在 GeometryFilter和下面的LabelPointReplacer之间添加另一个。

10、打开上面的Offsetter的属性，设置Y Offset为7.5。保持X和Z的值为0。
对于下面的Offsetter转换器进行同样的设置，只是把Y的值设置为-7.5。



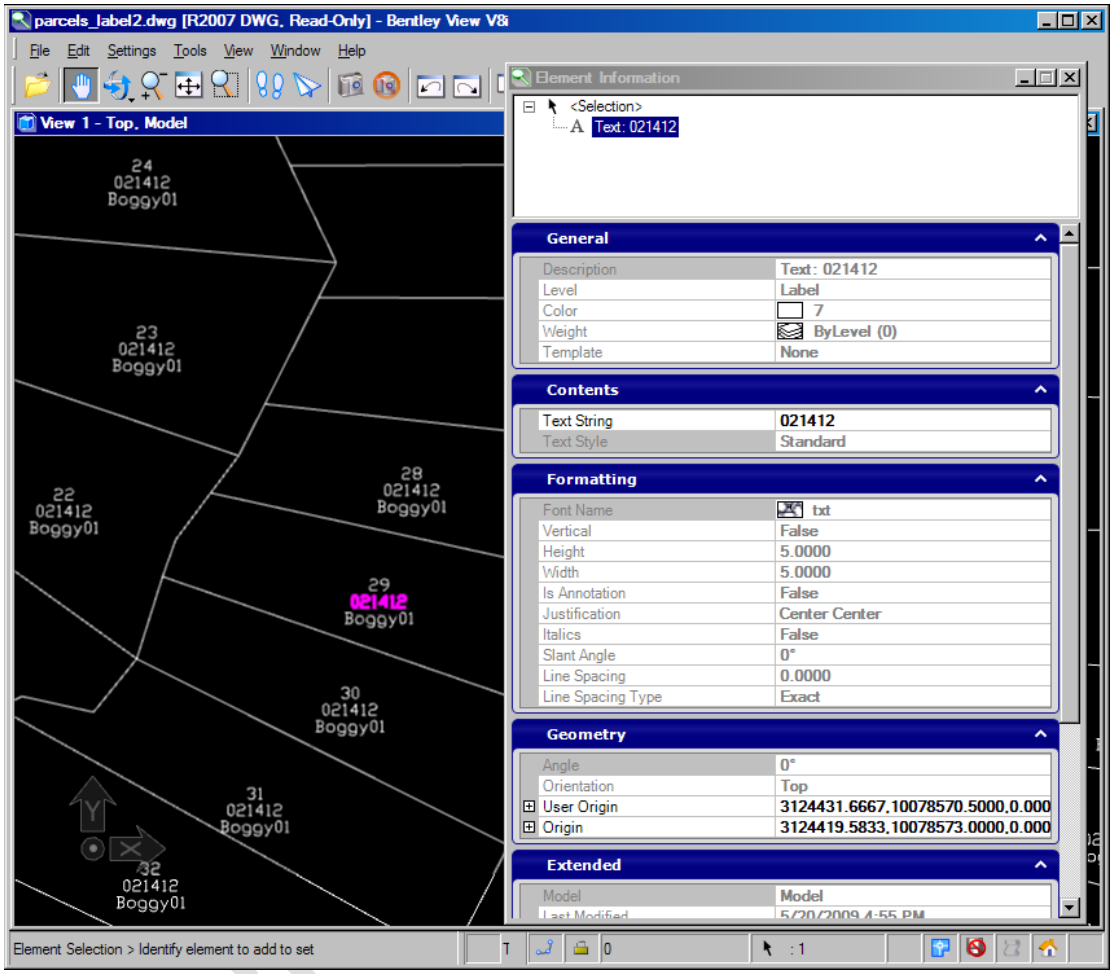
11、先前的标注以默认的底部左对齐放置，但是标注中心对齐的话会更好看些。

在LabelPointReplacers的右边添加一个AttributeCreator转换器，把他们全部连接起来。打开属性，添加一个属性autocad_justification，赋值为autocad_middle_middle。连接AttributeCreator到标注目标要素上。



12、设置输出数据集为Output\workshops\CAD\parcels_label2.dwg，运行这个工作空间。

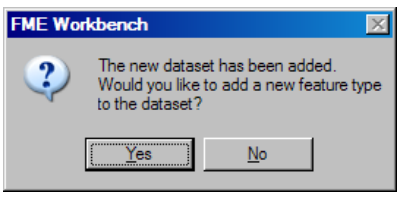
13、在Bentley View 中打开文件parcels_label2.dwg文件。



标注比之前有了很好的分离，有合适的大小，并且是希望的对齐方式。

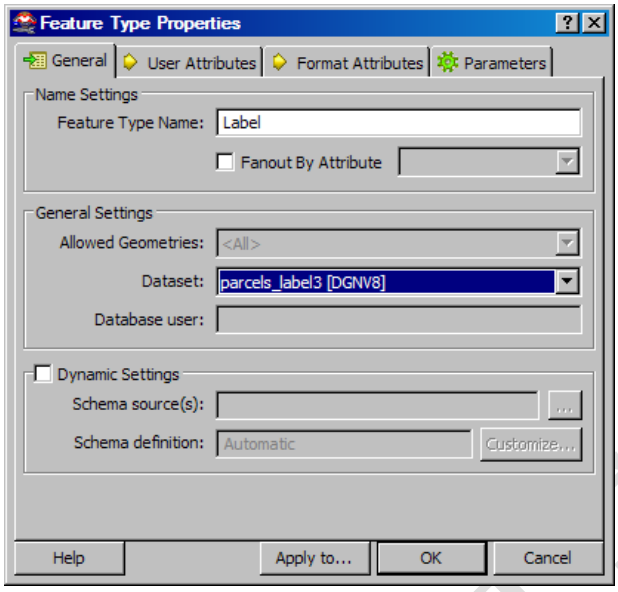
如果标注全部是一个单一元素，在CAD中编辑会更方便。在写入之前，我们可以聚合文本，不过这样写入的文本作为自定义块或单元，在文本编辑前需要展开。目前，FME不能写多文本到AutoCAD中，但是可以写入到MicroStation中。

14、在Workbench中打开之前的工作空间， 选择Destination Data – Add Dataset菜单。设置它的格式为 Bentley MicroStation Design (v8) ， 目标数据集设置为：Output\workshops\CAD\parcels_label3.dgn。



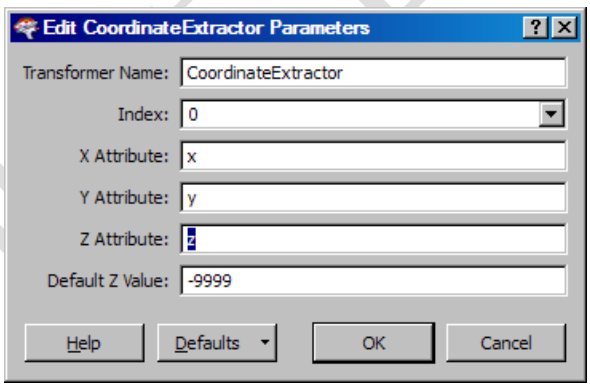
当弹出对话框，提示是否想添加一个新的要素类时，点击No。

15、打开Label目标要素类的属性，设置Dataset参数为parcels_label3 [DGNV8]。



16、对Parcels目标要素类进行相同的操作。现在所有的要素被写入到MicroStation 文件，而不是AutoCAD文件中。

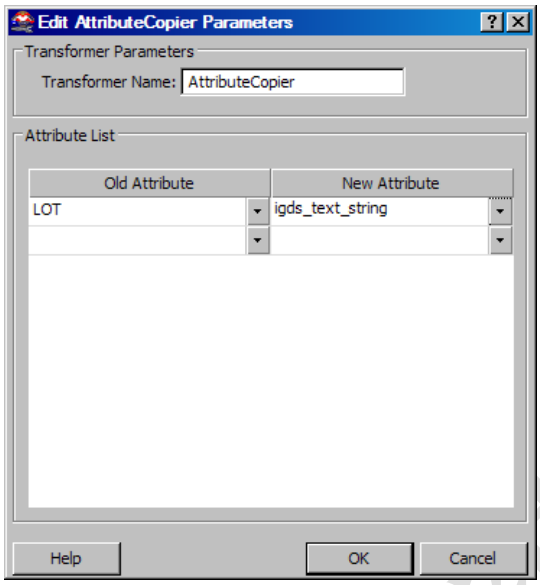
17、移除LabelPointReplacers和AttributeCreator转换器。添加一个CoordinateExtractor转换器，并且设置X Attribute为x、 Y Attribute 为 y 、 Z Attribute 为 z。



18、右键点击转换器，选择Duplicate来复制下这个转换器。重复创建另一个拷贝。

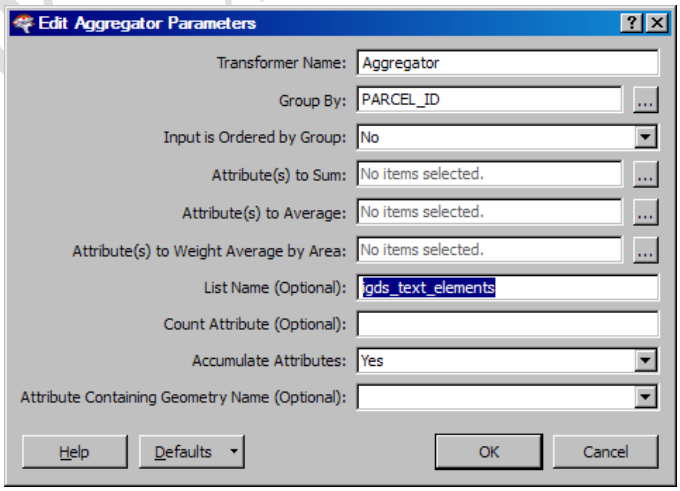
连接刚复制的一个CoordinateExtractor转换器到每个Offsetter，另一个直接连接到GeometryFilter的POINT输出端口。

19、我们已经移除了Labeler转换器，所以我们需要复制属性信息到文本串属性中。添加一个AttributeCopier，连接它到上面的CoordinateExtractor上。打开属性，复制LOT到igds_text_string上。



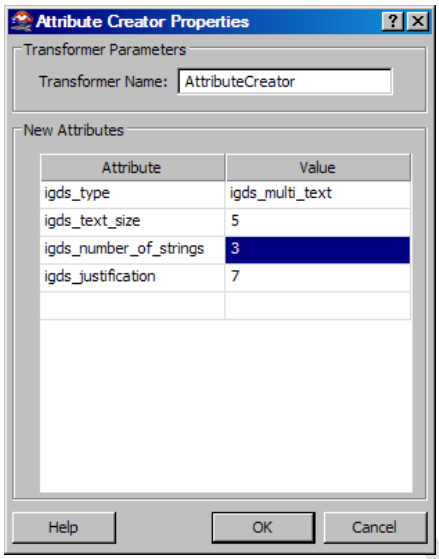
20、复制两个这个转换器，连接它们到另外的CoordinateExtractors上。打开属性，设置中间的AttributeCopier，复制PLAT 到igds_text_string。设置下面的AttributeCopier，复制SOURCE到igds_text_string。

21、现在我们定义了位置和文本字串名称，我们想连接所有的三个标注为一个节点。添加一个Aggregator，连接它到这三个AttributeCopy上。打开属性，设置Group By参数为PARCEL_ID，这对于每个标注是唯一的，设置List Name为igds_text_elements。

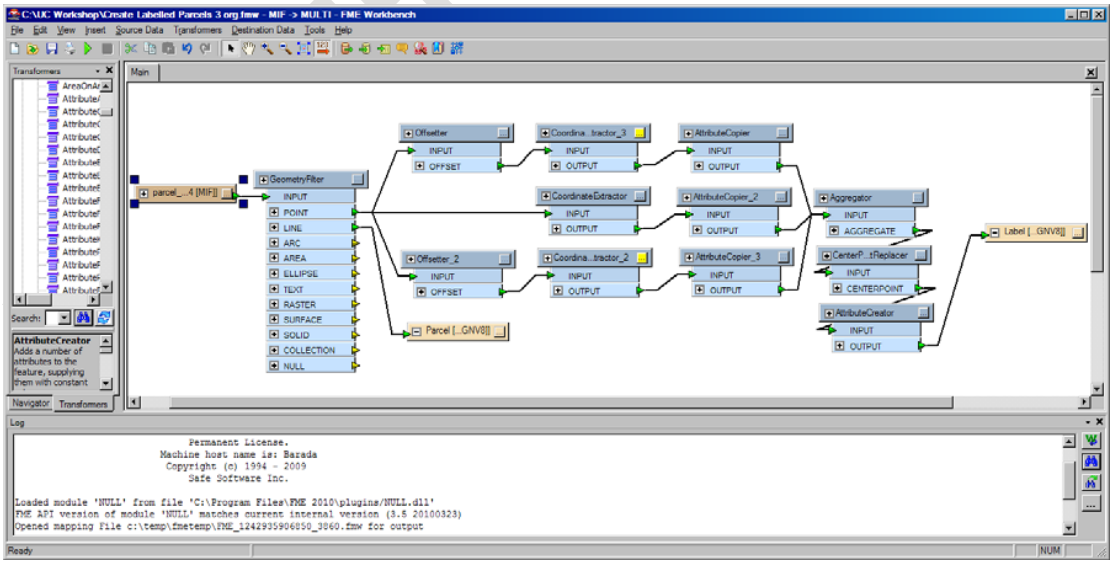


22、然而，我们不想写出一个聚类要素，所以我们需要用一个点来替换聚类几何。添加一个CenterPointReplacer ，连接它到Aggregator来把几何替换为一个点。

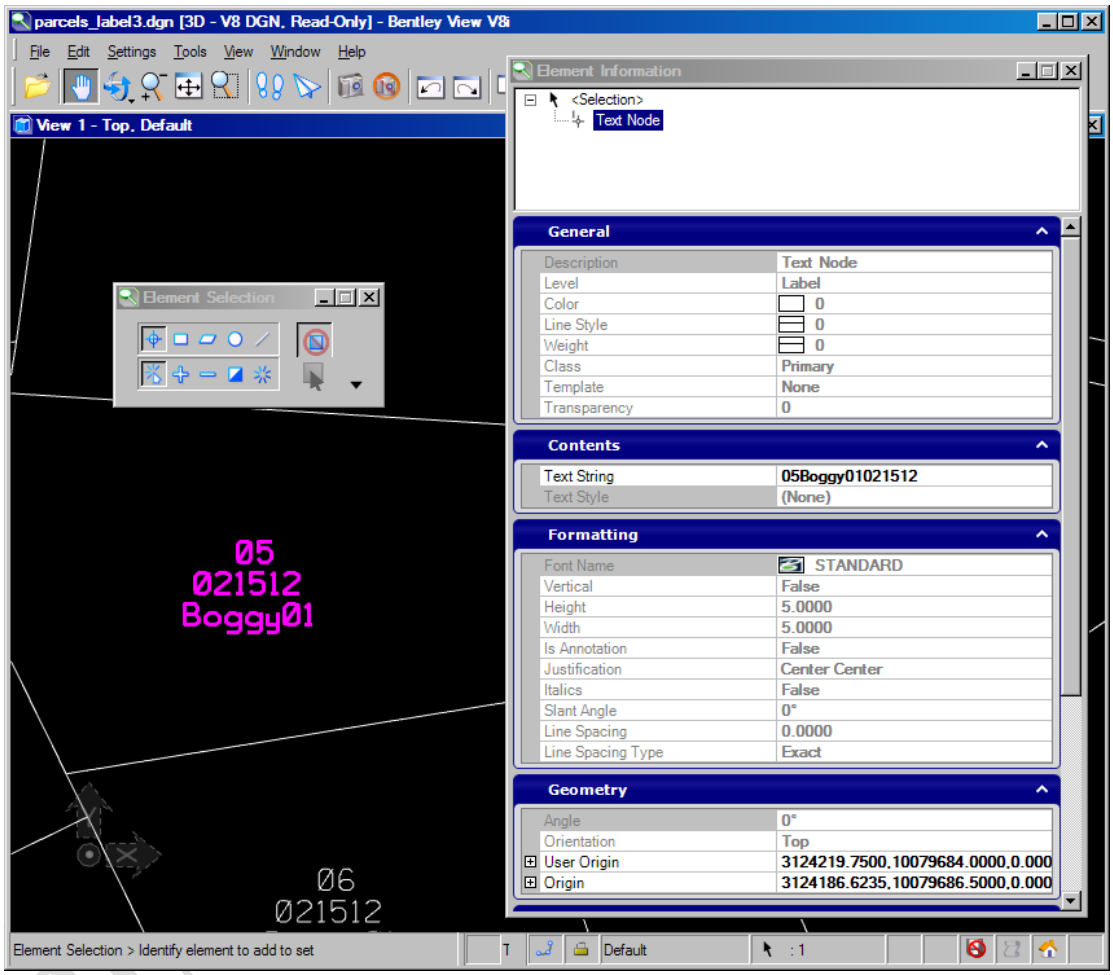
23、我们需要设置更多的属性来正确的写入复合文本。添加一个AttributeCreator ，打开属性。设置igds_type为igds_multi_text， igds_text_size 为5， igds_number_of_strings为3 ，igds_justification为7。



24、连接Label目标要素类到AttributeCreator ，保存工作空间为Create Labeled Parcels 3.fmw。



25、运行工作空间，在Bentley View中验证结果。

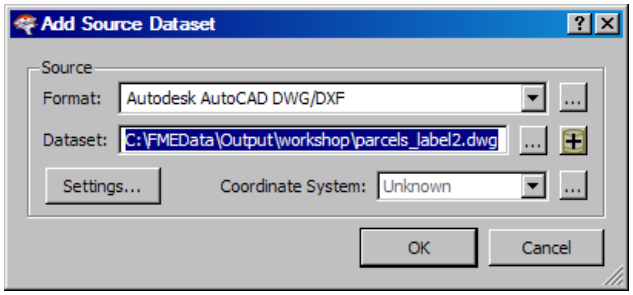


我们现在对于这三行文本只有一个标注

▪ 从面标注中提取（Extract from Area Labels）

创建面标注是相当简单的。从现有的标注中提取GIS属性有点复杂。我们将在parcels_label2.dwg 文件中提取标注保存到ESRI Access Geodatabase中。

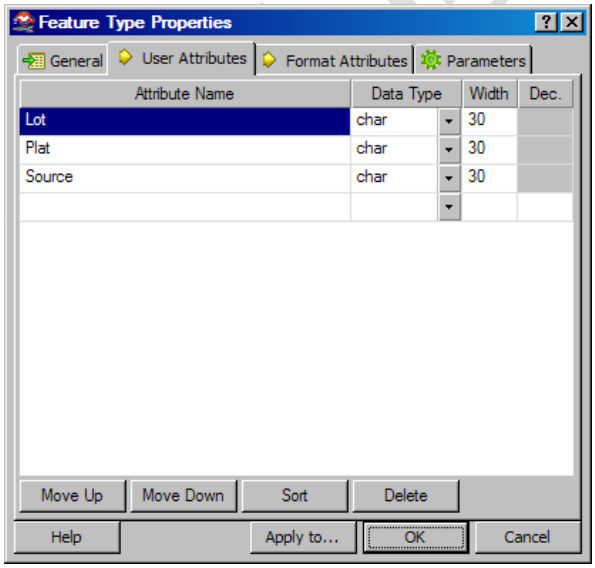
1、 打开Workbench，添加一个新的源数据集。选择Autodesk AutoCAD DWG/DXF格式作为源数据格式，选择Output\workshop\parcels_label2.dwg作为源数据集。



2、 添加一个新的目标数据集。选择ESRI Geodatabase (MDB) 作为目标数据集的格式，选择Output\workshop\parcels_geo.mdb 作为目标数据集。

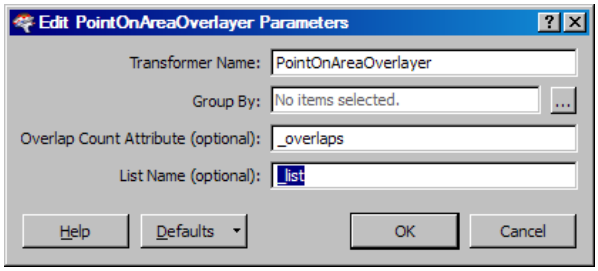
当询问你是否想添加一个新的要素类时，选择No。

3、 右键点击Parcel源要素类，选择Duplicate (As Destination)选项。打开结果中Parcel目标要素类的属性，设置运行的集合类型为geodb_polygon。选择User Attribute选项卡，创建用户属性Lot、Plat和Source。



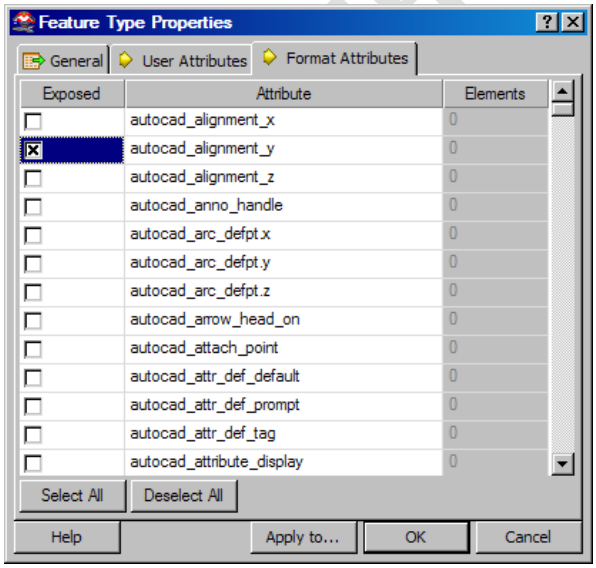
4、 因为地块输出为多边形，我们需要从Lot边界线中创建多边形。添加一个AreaBuilder 转换器连接到Parcels源要素类上。

5、 我们将覆盖多边形上的标注来转移标注内容，所以添加一个PointOnAreaoverlayer到工作空间。连接AreaBuilder的 AREA输出端口到PointOnAreaoverlayer的AREA输入端口，Label源要素类到POINT输入端口。打开转换器属性，设置List Name为_list。

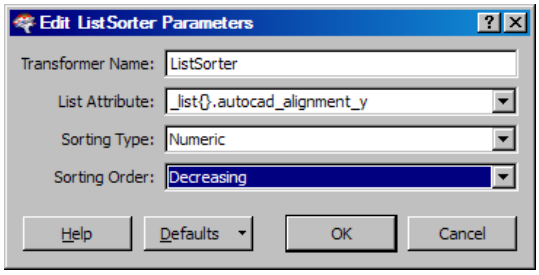


6、 现在我们已经把所有的标注附到了多边形上，但是我们不能从标注上提取信息，直到我们确定哪个标注相当于哪个属性。所以如果我们通过他们的Y坐标分类标注，可以知道我们放置Lot在最上面的LabelPointReplacer上，Plate放置在中间的上，Source放置在最下面一个上；我们知道它们的顺序是Lot、Plat、Source。幸运的，AutoCAD Reader向我们提供了这个标注的一个包含Y坐标的属性。

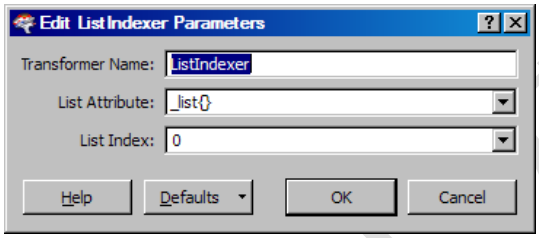
7、打开标注源要素的属性，选择Format Attributes选项卡。
选中autocad_alignment_x、autocad_alignment_y 和autocad_text_string选项，在Workbench中暴露这些属性。



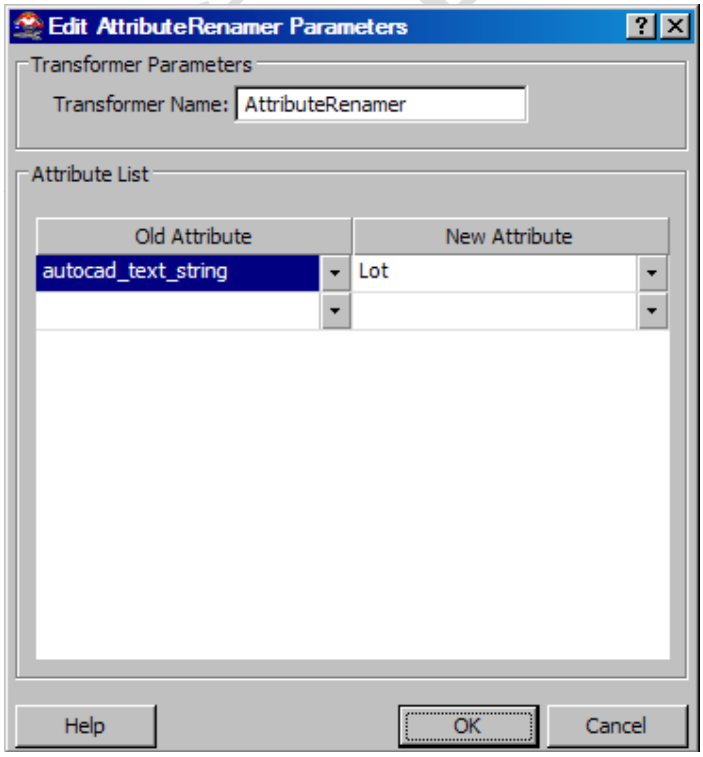
8、添加一个ListSorter到工作空间，连接它到PointOnAreOverlayer的AREA输出端口。
打开属性，设置List Attribute为_list{}.autocad_alignment_y。设置Sorting Type为Numeric ，设置Sorting Order为Decreasing。



9、现在这个列表从顶到底来分类，我们想提取属性值。添加一个ListIndexer，连接它到ListSorter，设置List attribute 为_list{} ，设置List Index 为0。



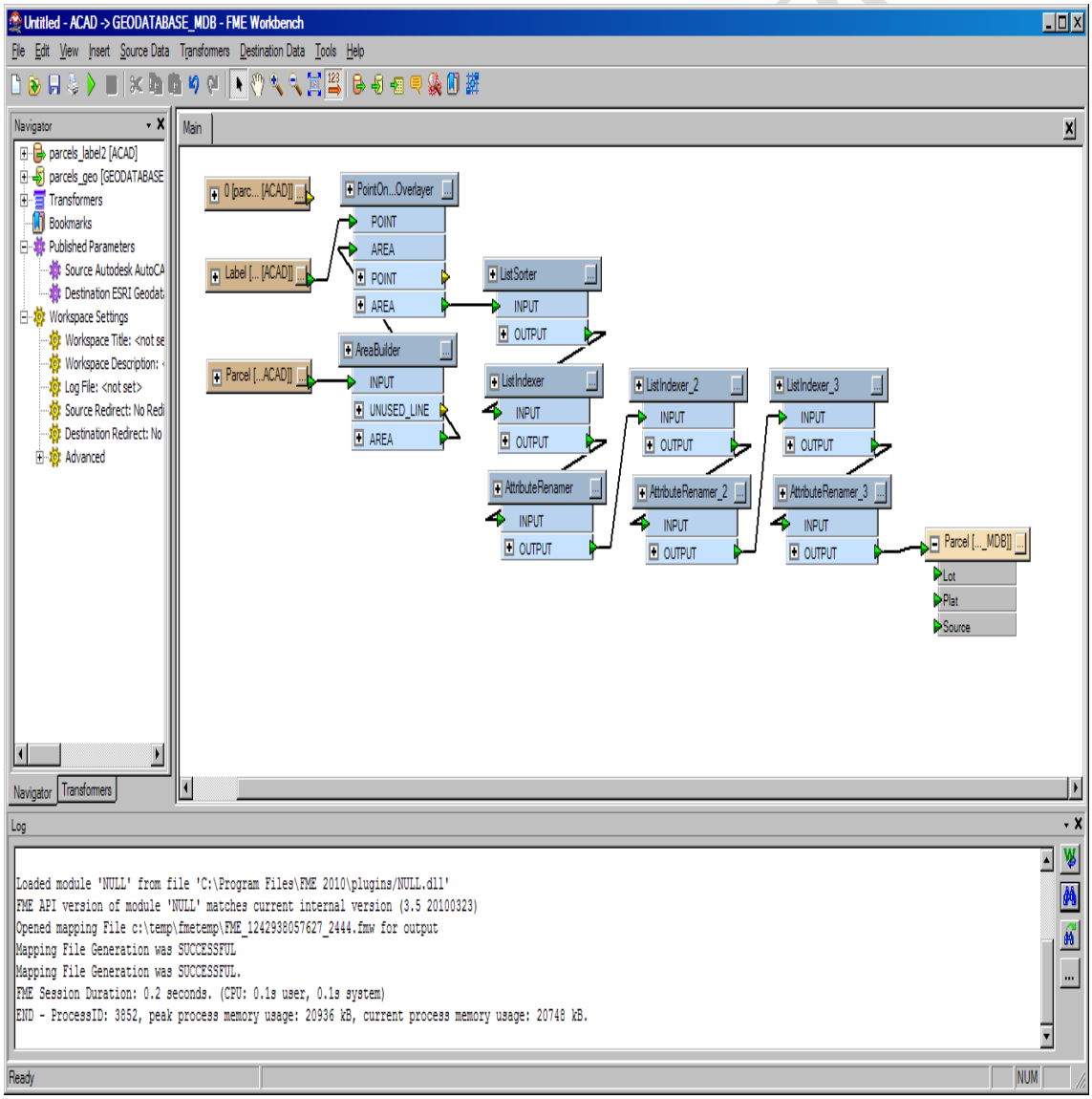
10、我们将获取列表属性_list{0}.autocad_text_string，提交它到属性autocad_text_string。
添加一个AttributeRenamer ， 重命名autocad_text_string为Lot，连接它到ListIndexer上。



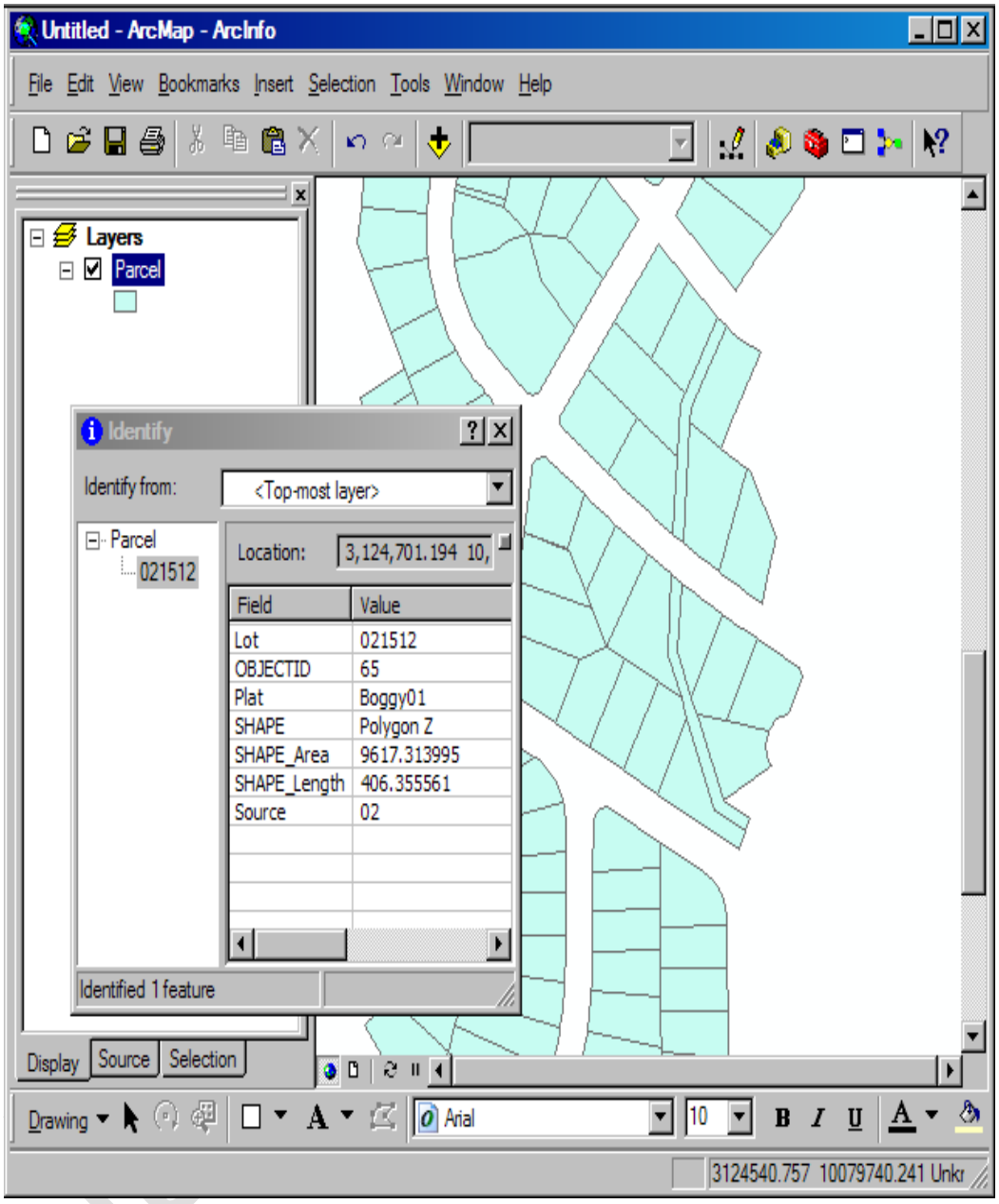
11、复制两个ListIndexer 和AttributeRenamer转换器，按ListIndexer、AttributeRenamer、ListIndexer和AttributeRenamer次序把它们连接起来。

设置第二个ListIndexer的List Index 为1，第三个ListIndexer的List Index 为2。设置第二个AttributeRenamer， 重命名autocad_text_string 为 Plat，第三个AttributeRenamer， 重命名autocad_text_string 为Source。

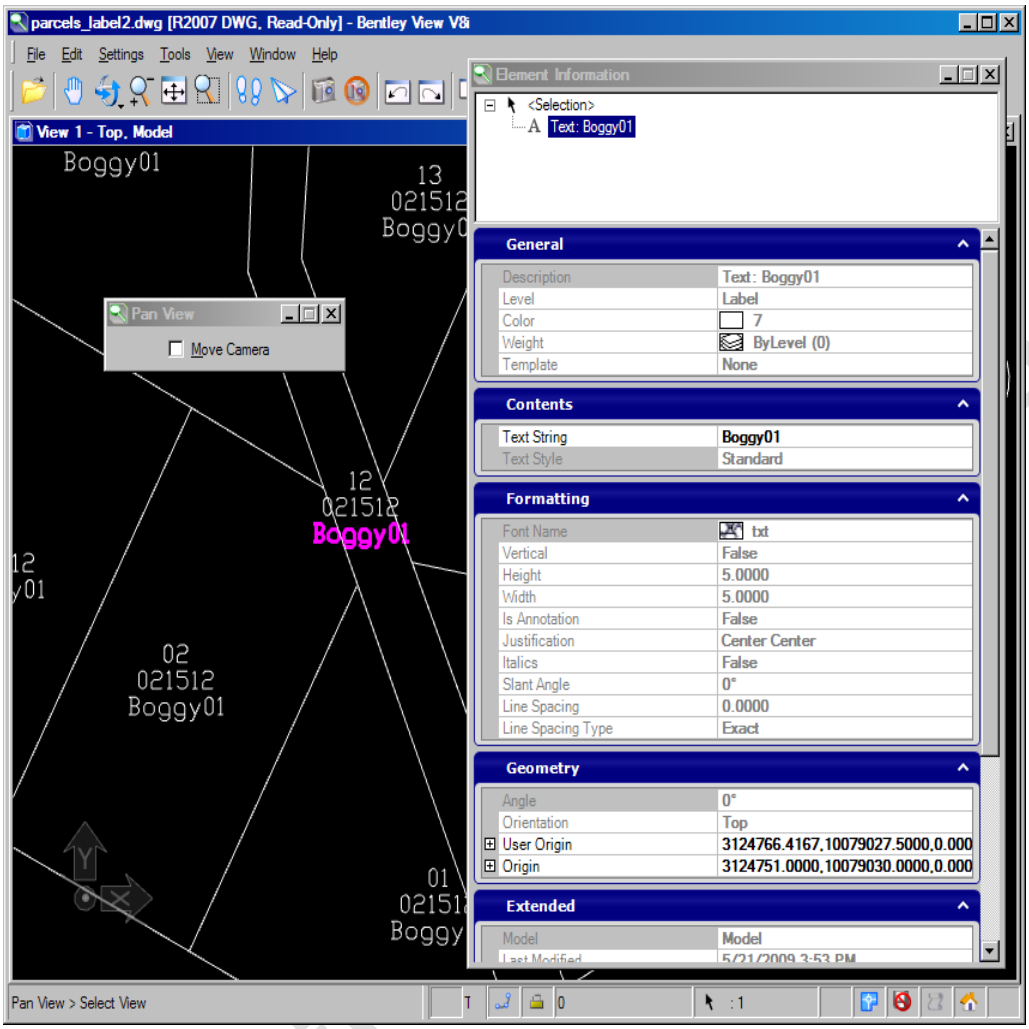
12、连接Parcels目标要素类到最后一个AttributeRenamer上。



13、保存工作空间为Extract Area Label 1，然后运行这个转换。在ArcMap中打开parcel_geo.mdb来检验属性。



多数多边形有正确的属性，不过在lot 02中，它们看起来出现了错误。让我们查看下源数据看看是否能找出原因。



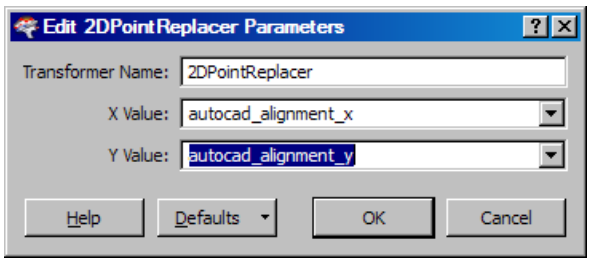
Lot 12的文本看起来覆盖在了lot 02上，但是因为文本放置在了地块的质心上，应该认为是多边形被覆盖的位置。

然而，在AutoCAD和MicroStation中有一个巧合——即使文本调整后放置在地块的质心上，文本的位置在DWG文件中存储的是文本的左下方。这种情况下，添加超过两个的标注到lot 02上将会有错误的顺序。Bentley View把作为源（存储位置）和用户源（放置位置）的两个位置都显示出来了。

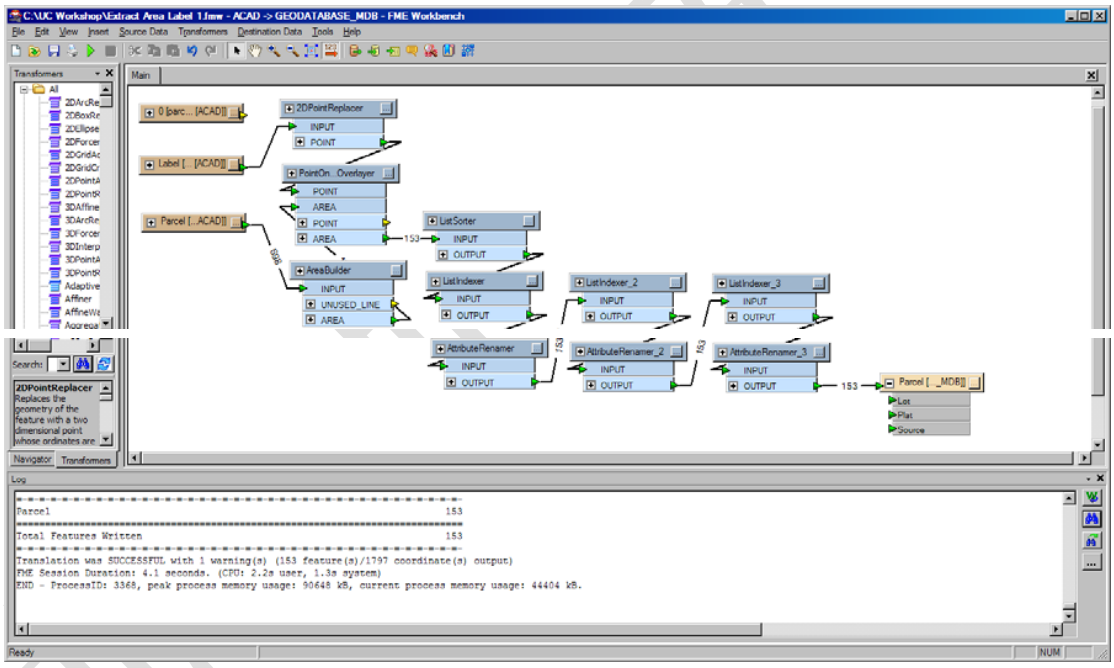
FME 读入存储位置作为文本的放置点，但是它也计算放置位置，并把它存储到属性autocad_alignment_x、 autocad_alignment_y 和autocad_alignment_z（在MicroStation 中是igds_insertion_x 和igds_insertion_y）中。

在一个2DPointReplacer转换器中使用这些计算值，我们可以替换标注源数据的位置。

14、在源标注要素类和PointOnAreaOverlayer之间，插入一个2DPointReplacer到工作空间。打开属性，设置X Value 为autocad_alignment_x，设置 Y Value为autocad_alignment_y。



15、关闭ArcMap，保存工作空间然后再次运行转换。



16、在ArcMap中打开Geodatabase，检查有问题的地块——现在它看起来应该有了正确的属性。像把标注写入地块属性一样，我们或许想把属性作为标注写入Geodatabase。

17、添加一个LabelPointReplacer，连接它到2DPointReplacer的POINT输出端口来把点转换回标注。设置Label Attribute为autocad_text_string， Label Height为5。

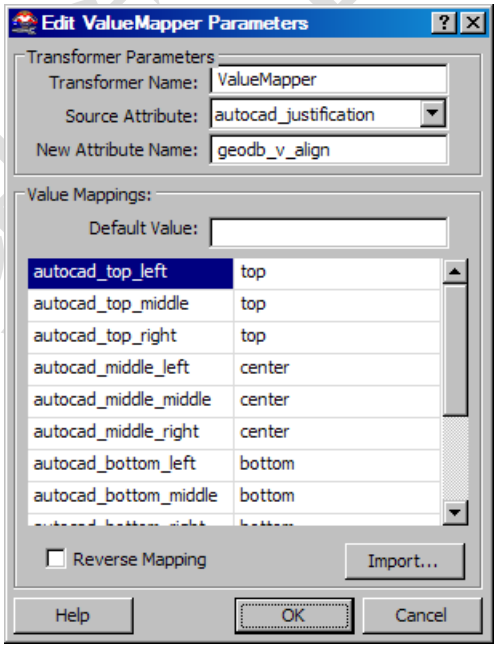
通过通用（generic）属性，FME自动从一种格式到另一种格式来绘制文本字串和文本高度，因为在所有的格式间都有相应的标准。不过在文本对齐方式间没有标准，所以我们必须手动的绘制。为了使这更容易，Safe公司创建了一个网页来详细的说明大多数常见格式间的文本对齐的表示(http://www.fmepedia.com/index.php/Justification)。我们将利用这个网页来表示AutoCAD文本对齐方式到相应的Geodatabase中。

18、在这个网页上， 你会看到AutoCAD只有一个对齐属性，而Geodatabase中有两个，所以我们需要使用两个ValueMappers。

打开Label源要素类属性，暴露autocad_justification属性。添加一个ValueMapper转换器，把它连接到LabelPointReplacer上。

设置Source Attribute为autocad_justification，设置New Attribute Name为geodb_v_align，给ValueMapper添加下列值：

autocad_top_left	top
autocad_top_middle	top
autocad_top_right	top
autocad_middle_left	center
autocad_middle_middle	center
autocad_middle_right	center
autocad_bottom_left	bottom
autocad_bottom_middle	bottom
autocad_bottom_right	bottom
autocad_baseline_left	baseline
autocad_baseline_middle	baseline
autocad_baseline_right	baseline

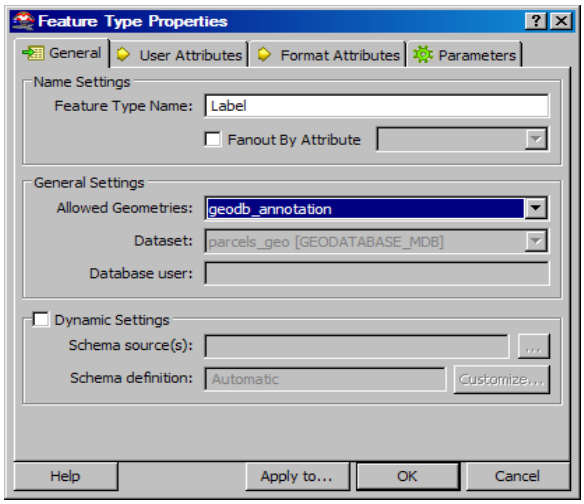


19、再添加一个ValueMapper，把它与第一个ValueMapper连接起来。

设置Source Attribute为autocad_justification，设置New Attribute Name为geodb_h_align，在ValueMapper上添列下列属性：

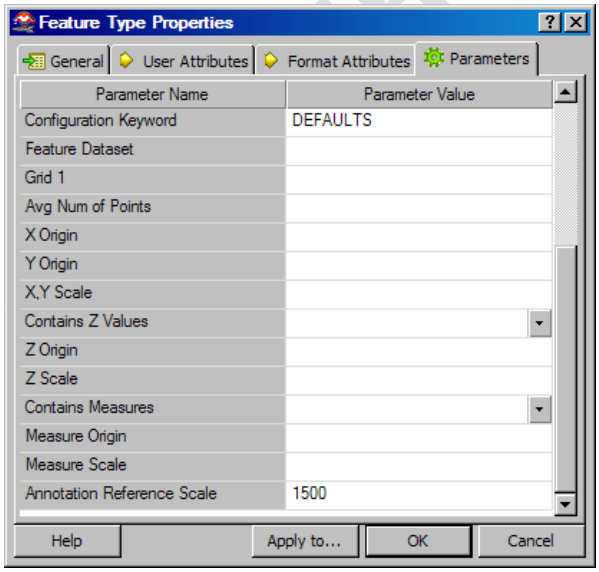
autocad_top_left	left
autocad_top_middle	center
autocad_top_right	right
autocad_middle_left	left
autocad_middle_middle	center
autocad_middle_right	right
autocad_bottom_left	left
autocad_bottom_middle	center
autocad_bottom_right	right
autocad_baseline_left	left
autocad_baseline_middle	center
autocad_baseline_right	right

20、右键点击工作空间区，插入目标要素。命名这个要素为Lables，设置Allowed Geometries为geodb_annotation。

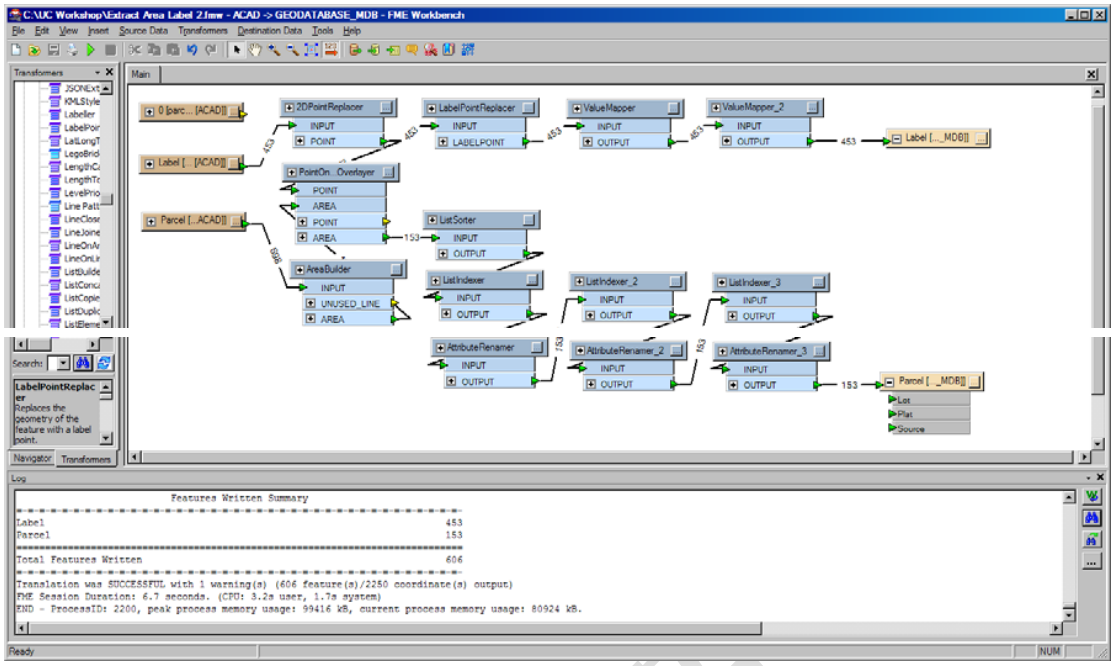


在AutoCAD中和FME内部，文本高度以它正使用的单位计算，在我们的例子中是米。在Geodatabase中，文本高度以显示点计算，所以在所有的缩放级别上，文本仍保持相同的大小。FME会转换高度单位，但为了更好的效果，我们需要设置参考尺度——在这个比例上显示的高度与地面高度匹配。

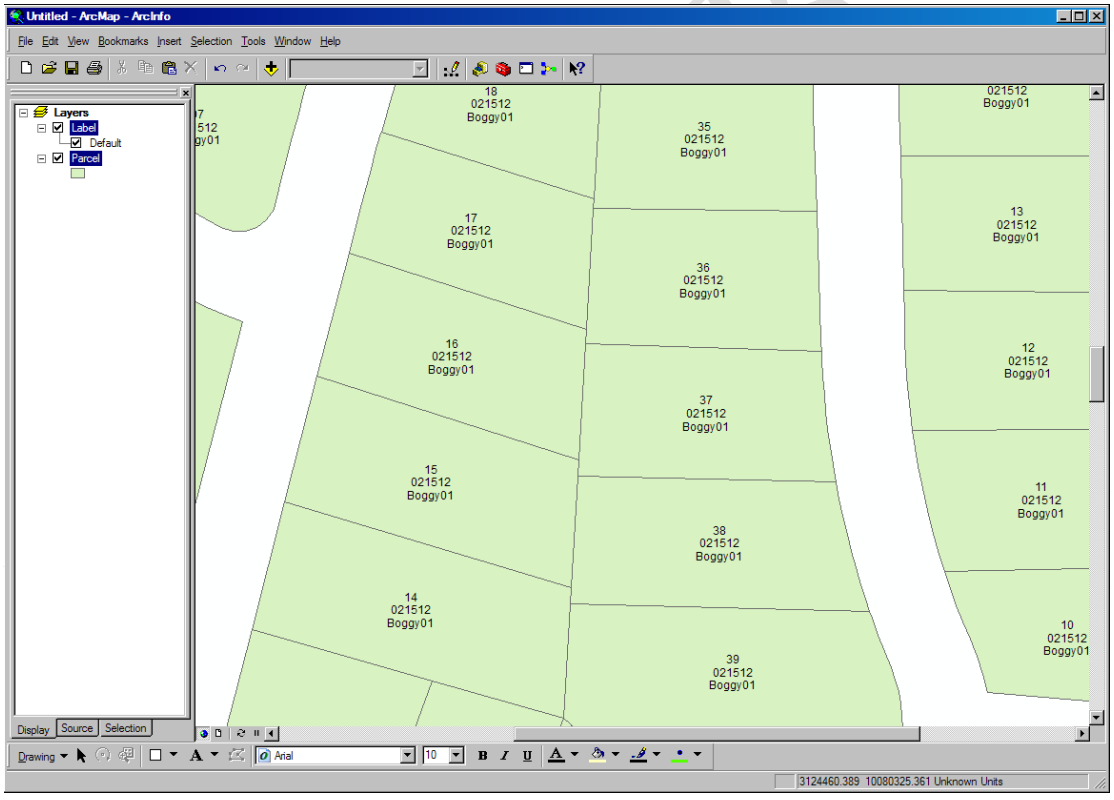
21、切换到Parameters选项卡，向下滚动到最后，设置Annotation Reference Scale为1500。



22、连接Label目标要素类到第二个ValueMapper。保存工作空间，命名为Extract Area Label 2.fmw。



23、在ArcMap中打开Geodatabase，验证结果。



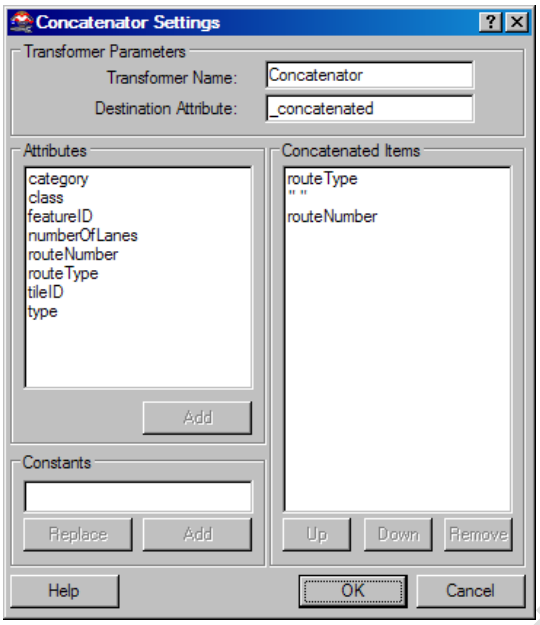
创建线性标注

在这个练习中我们提取主要的道路，并用这些路的类型和号码来标注它们。

1、在Workbench中打开一个新的工作空间，添加源数据集：C:\FMEData\Data\Roads\RoadLine.mif。添加一个Autodesk AutoCAD DWG/DXF格式的目标数据集，设置目标数据集为： C:\FMEData\Output\workshop\roads.dwg。

2、右键点击Workbench工作台，插入目标要素Labels。执行相同的操作添加目标要素Roads。

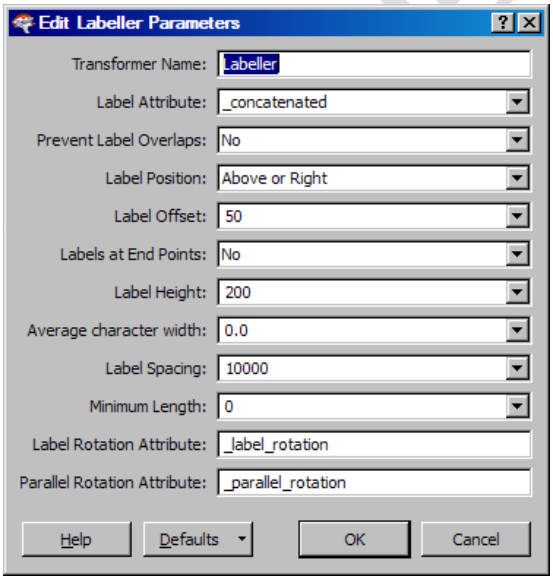
3、我们想连接routeType 和routeNumber 为一个属性来做标注，所以添加一个Concatenator， 并连接它到RoadLine要素类。添加routeType 和 routeNumber 属性，用一个空格常量把它们分隔开。



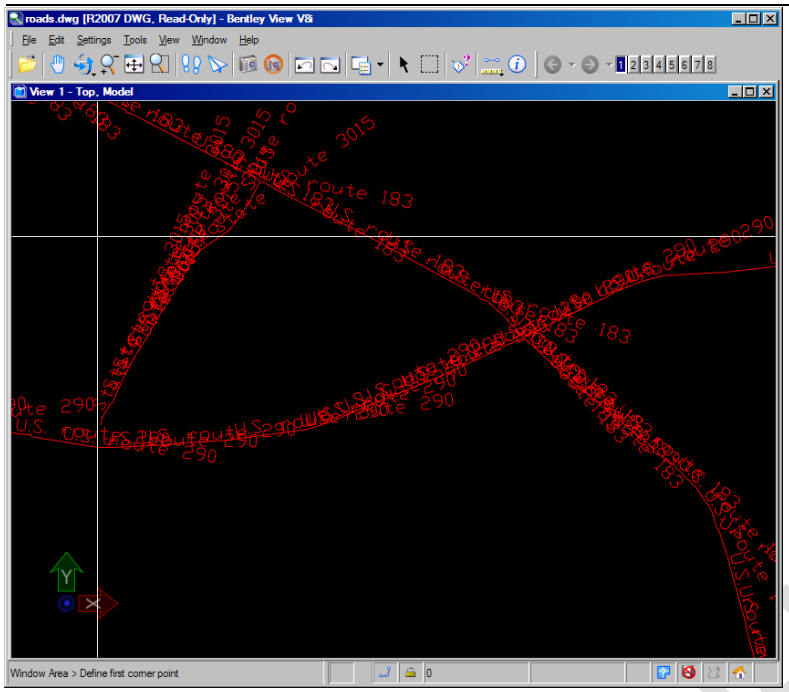
4、考虑到输出，我们只想保留主要道路，给他们一个有效的标注。为了达到这个目的，我们添加一个Tester来检查这个属性不只是包含一个空格的new _concatenated属性。

连接Tester 到Concatenator，然后打开Tester的属性，设置左边的值为_concatenated，设置Operator 为!=，设置右边的值为一个空格。连接Roads目标要素类到Tester 的PASSED端口。

5、添加一个Labeller转换器来标注道路，连接它到Tester 的PASSED 输出端口。设置Labeller 的参数如下图所示：



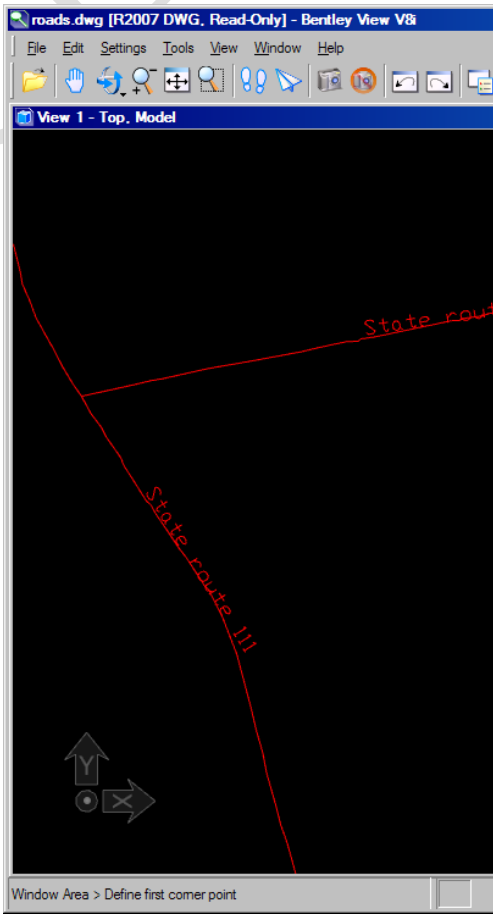
6、连接Labels目标要素类到Labeller 的LABEL端口，保存工作空间，命名为Create Labeled Lines.fmw。运行工作空间，在Bentley View中检验输出文件。



道路被分成了许多小片，每个小片都有它们自己的标注。

这不是我们想看到的输出结果，所以我们将在标注道路之前先把道路连接在一起。

7、添加一个LineJoiner，把它插入到Tester和Labeller之间。设置Group By attribute为_concatenated，然后重新运行这个工作空间。在Bentley View中查看结果。



这次得到了更好的结果。

你可以调整Labeller的参数来改变标注的大小、位移和频率。

从线性标注中提取

从多边形里的标注中提取信息是非常简单的，因为标注或者在多边形的内部，或者不在内部。线性标注更有趣，因为我们将根据邻近关系来把标注和线关联起来。

1、打开Workbench，创建一个工作空间。添加源数据集C:\FMEData\Data\Roads\MajorRoads.dgn，设置为Bentley MicroStation Design (V8)格式。

在源要素对话框中，点击Settings按钮，设置Group Elements By 为Level Name。

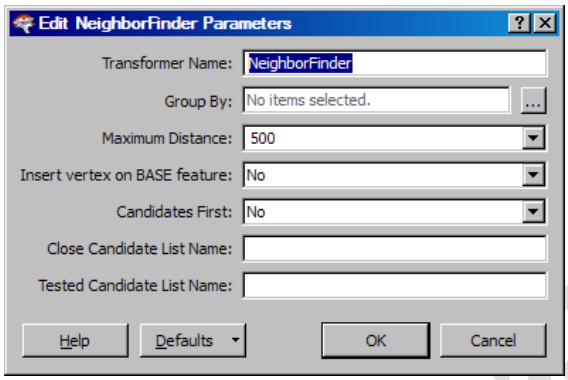
2、添加一个新的目标要素类，格式为ESRI Geodatabase (MDB)，设置目标数据集为：C:\FMEData\Output\workshop\roads_geo.mdb。

添加一个新的目标要素类Roads，同时添加用户属性Label。

3、打开Labels源要素类的属性，暴露格式属性igds_text_string。添加一个AttributeCopier ，并连接它到Labels要素类。设置它来拷贝igds_text_string属性到Label上。

4、为了把标注和道路匹配起来，我们将添加一个NeighborFinder转换器。

连接NeighborFinder的BASE端口到Roads要素类，CANDIDATE 端口到AttributeCopier。这个转换器将把过滤距离内的所有最近的标注添加到道路线上。分别连接MATCHED和UNMATCHED_BASE 端口到目标Roads要素类。打开NeighborFinder属性，设置参数值如下：



5、保存工作空间为Extract Line Labels 1.fmw，并运行它。在ArcMap中打开roads_geo.mdb 来验证结果。我们要注意的是，最左边的南北路当它向南伸展时，更改了名字。东西方向上与它相交的路都把名称更改为51st ST E。

如果我们在Bentley View中打开Major Roads.dgn，我们会看到引起这个问题的原因。51st ST E 标注比正确的标注距离这些路更近。看起来我们把标注和道路匹配起来不仅仅要考虑距离。

如果我们注意下这些标注，它们的转动方向大体上与它们标注的路段方向匹配。让我们也添加一个方向来过滤工作空间。

6、打开NeighborFinder属性，设置 Close Candidates List Name 为_list。现在代替单一的最接近的标注，NeighborFinder 会返回最小距离内的所有标注的列表。

右键点击NeighborFinder 的MATCHED 端口，选择Visualizer来连接Visualizer。运行这个工作空间。

Viewer会打开，并且显示匹配的道路。选择菜单上的①图标，点击道路线。如果你看到信息窗口，你会看到_list{} 有一个或更多的元素和许多子属性。我们感兴趣的两个属性是_list{}.distance和and _list{}.angle。距离是线上最近的点到标注的距离，角度是从线上最近的一点到标注的角度，有效的是在这一点上的线的角度是90度。这正是我们将要与标注方向对比的角度。

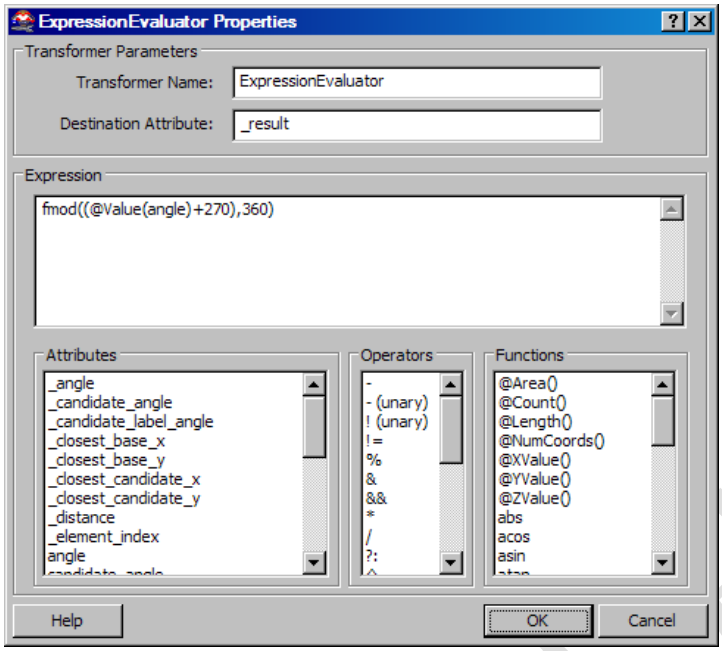
7、第一步是打散列表，以便我们能把他们进行匹配。为了保持匹配/道路关系的轨迹，我们将对每个路段创建ID号。在Road源要素类和NeighborFinder之间插入一个Counter。设置Counter Output Attribute为ID。

8、把输出端口MATCHED 和 UNMATCHED输出端口与Roads目标要素类的连接。

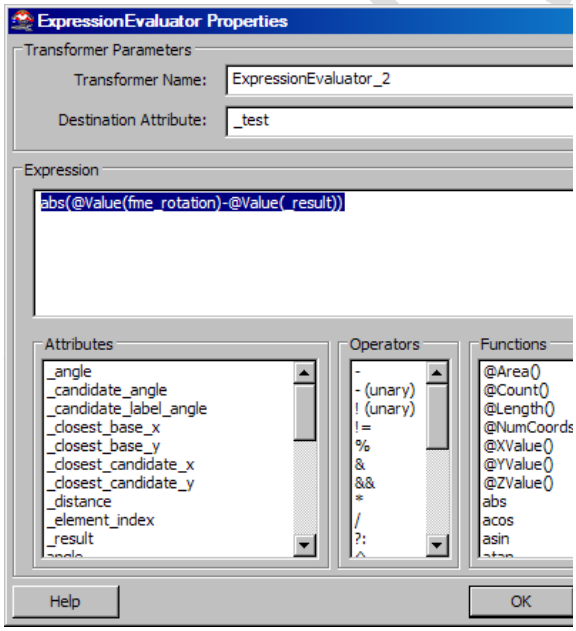
添加一个ListExploder 连接它到NeighborFinder 的MATCHED端口。设置List Attribute为 _list{}。

9、为了把路段的角度属性旋转90度，我们添加一个ExpressionEvaluator ，设置表达

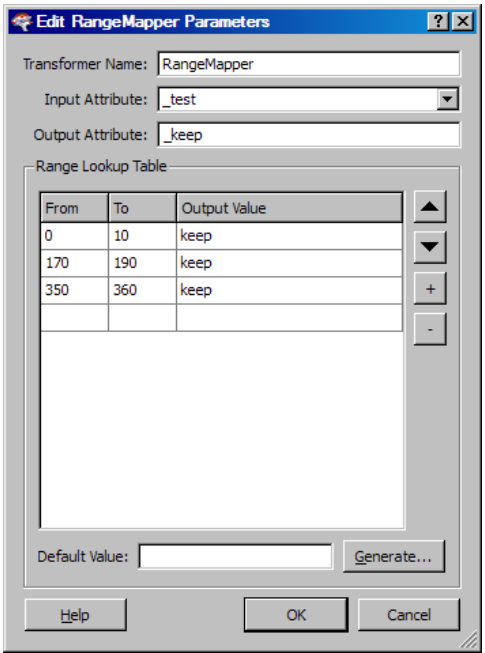
式为：fmod((@Value(angle)+270),360)。这将把这个角度增加270度，然后除以360度，保留余数。保留 Destination Attribute 为_result。



10、打开Labels源要素的属性，暴露属性fme_rotation属性，以便我们可以看到标注的旋转。添加另一个ExpressionEvaluator，把它与第一个ExpressionEvaluator连接起来。设置expression为abs(@Value(fme_rotation)-@Value(_result))。这会计算标注角度和路段角度的差值。



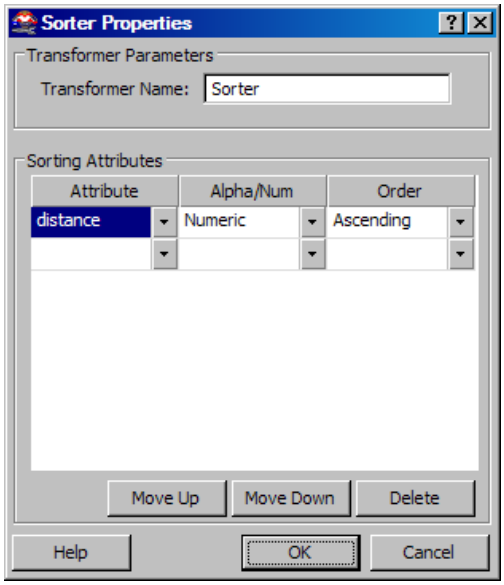
11、只保留在线上的标注，并且我们需要找出在容差范围内，标注的方向与线的方向相同或相反的线段。容差是10度意味着_test 值为0 - 10, 170 - 190, and 350 - 360 是可接受的。对于FME 2010，引入了一个新的转换器RangeMapper是个好时机。添加RangeMapper并连接它到第二个ExpressionEvaluator。打开属性，设置Input



Attribute为 _test，设置Output Attribute 为_keep。添加范围0 - 10, 170 - 180和350 - 360 ，把 Output value都设为keep。

12、为了丢弃不成功的标注，添加一个Tester ，测试条件为_keep = keep ，把它连接到RangeMapper上。

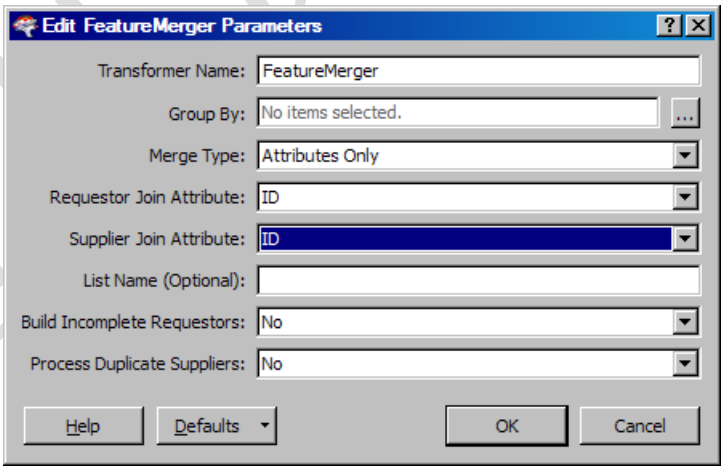
现在我们只保留了路的在线上的标注，我们想再次找出最近的标注。我们已经知道到线的距离，我们只需根据距离来把线分类，清除除第一个之外的所有标注。



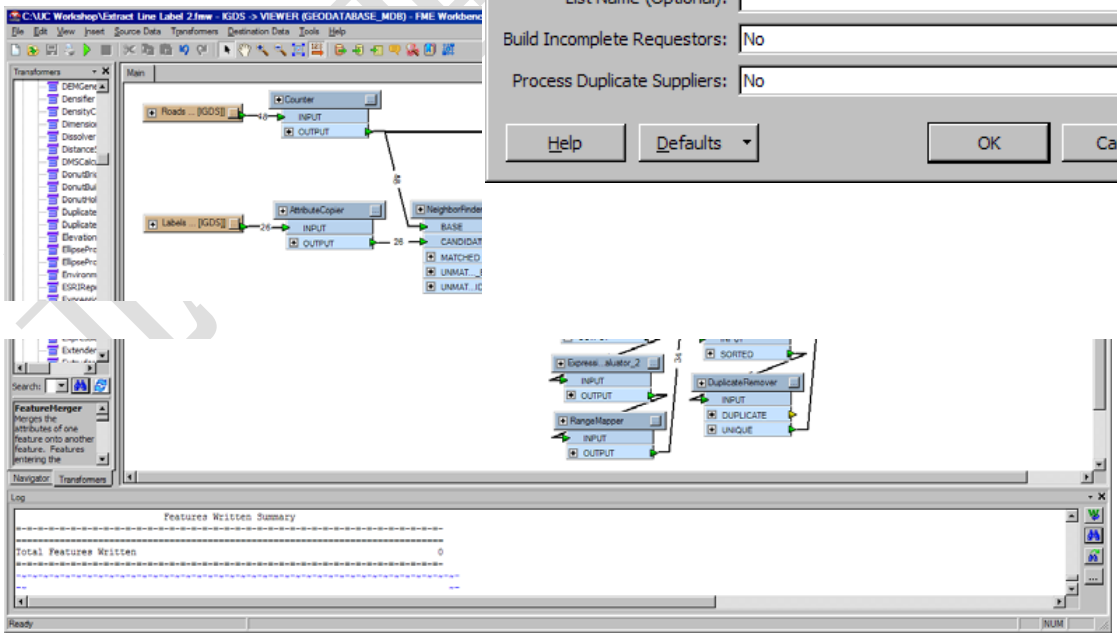
13 、 添 加 一 个 Sorter ， 设 置 Sorting Attribute 依 次 为 _distance 、 Numeric 、 Ascending。

14. 添加一个 DuplicateRemover ，并设置 Key Attribute 为ID。 UNIQUE的端口将输出距离线最近的标注，不过我们仍需转移标注到道路线上。

15 、 添 加 一 个 FeatureMerge ， 连 接 DuplicateRemover的UNIQUE到Supplier端口，连接 DuplicateRemover 的 REQUESTOR 端 口 到 Counter上。分别打开Supplier 和Requestor属性，把Join Attribute都设置为ID。



16、把FeatureMerger 的COMPLETE和INCOMPLETE端口都连接到Roads目标要素。保存工作空间为Extract Line Label 2.fmw。



17、运行工作空间，在ArcMap中检查Geodatabase。不是所有的道路段都被标注，但是，没有错误的标注。你可以调整最大距离和角度的容差来得到更好的结果。

用符号分类属性

GIS属性包含了被枚举的值（行车路线的号码、河流类型等），在CAD中经常绘制到线符号中，因为它提供了一个简单的方法转换信息到地图用户，没有使用标注那样散乱。



创建符号

在这个练习中，我们将使用ValueMappers和一个外部的表来绘制GIS属性到CAD符号中。

1、打开Workbench，创建一个新的工作空间。

2、添加C:\FMEData\Data\Roads\RoadLine.mif作为一个新的Mapinfo MIF/MID 源数据集。

3、添加一个新的MicroStation (V8) 格式的目标数据集C:\FMEData\Output\workshop\roads.dgn。

我们将执行下面的映射关系：

numberOfLanes	igds_weight
-9999	1
2	2
3	3
4	4
5	5
6	6

class	igds_color
class 1	1
class 2	2
class 3	3
class 4	4
unknown	5

type	igds_style
Primary route	0
Secondary route	1
Road	2
Ramp in interchange	3
Underpassing	4
Arbitrary line extension [Code Deleted 07/95]	5
Coincident feature [Code Deleted 10/93 memo (05/94 Standard)]	6

为了完成前两个映射，我们将使用一个ValueMappers，而为了完成最后一个绘图，我们将使用一个外部的CSV表。

4、添加两个ValueMappers，一个连在另一个的后面，使用上面详细的映射要求来创建它们。

5、打开记事本，创建一个对于igds_style绘图的用逗号分割的文件：

```
igds_style,type
0,Primary route
1, Secondary route
2, Road
3, Ramp in interchange
4, Underpassing
```

北京世纪安图

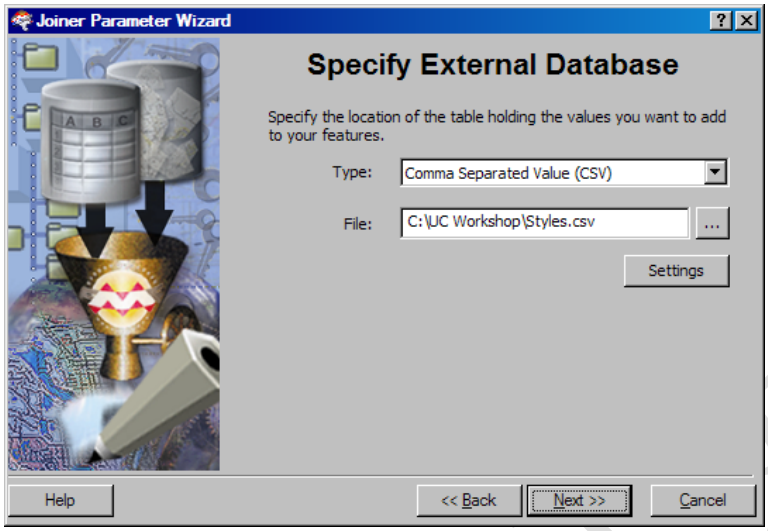
5, rbitrary line extension [Code Deleted 07/95]

6, Coincident feature [Code Deleted 10/93 memo (05/94 Standard)]

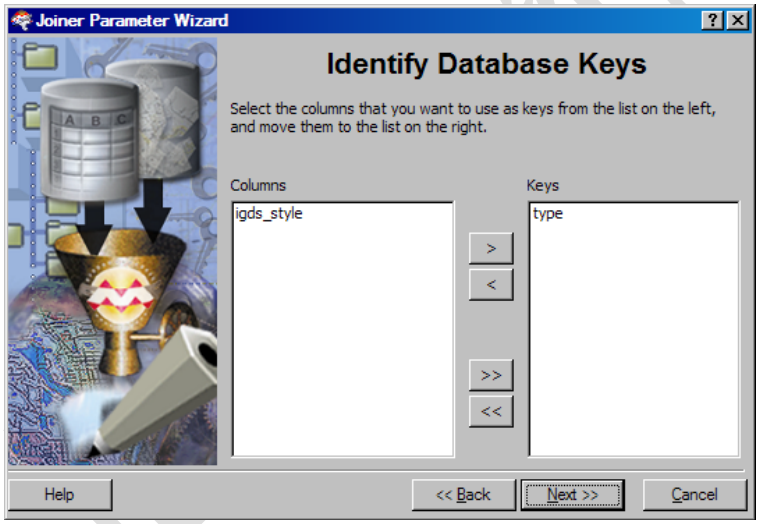
6、保存这个文本文件为Styles.csv。

7、添加一个Joiner 到工作空间，连接它到上一个ValueMapper上。

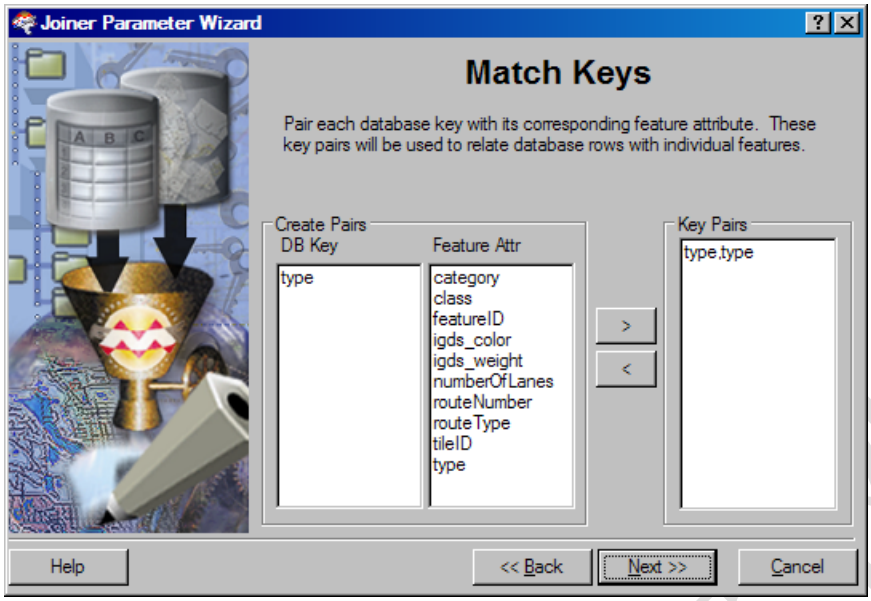
打开属性来启动一个Joiner向导。选择Comma Separated Value (CSV)来设置Type，选择你刚刚创建的csv 文件来设置File。



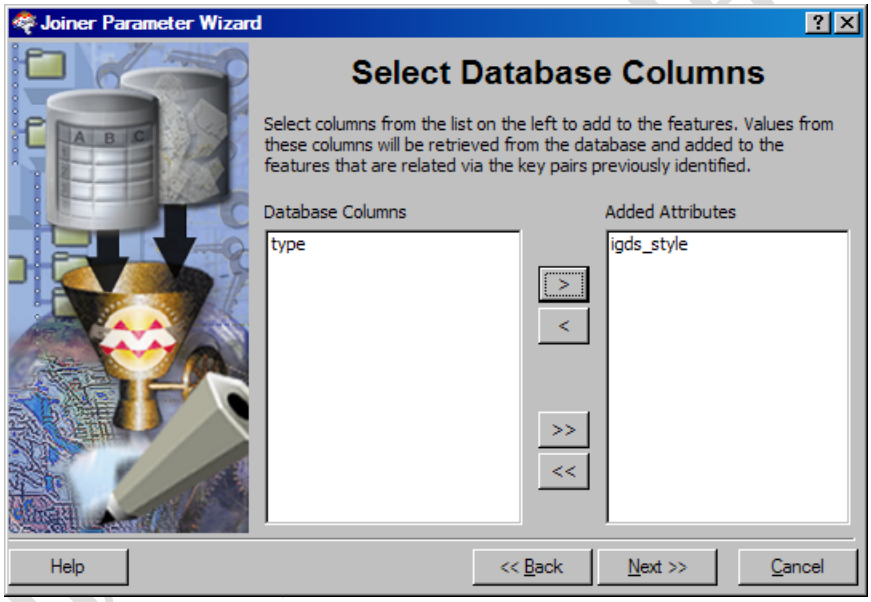
8、在下一个屏幕上，选择type作为数据库关键字。



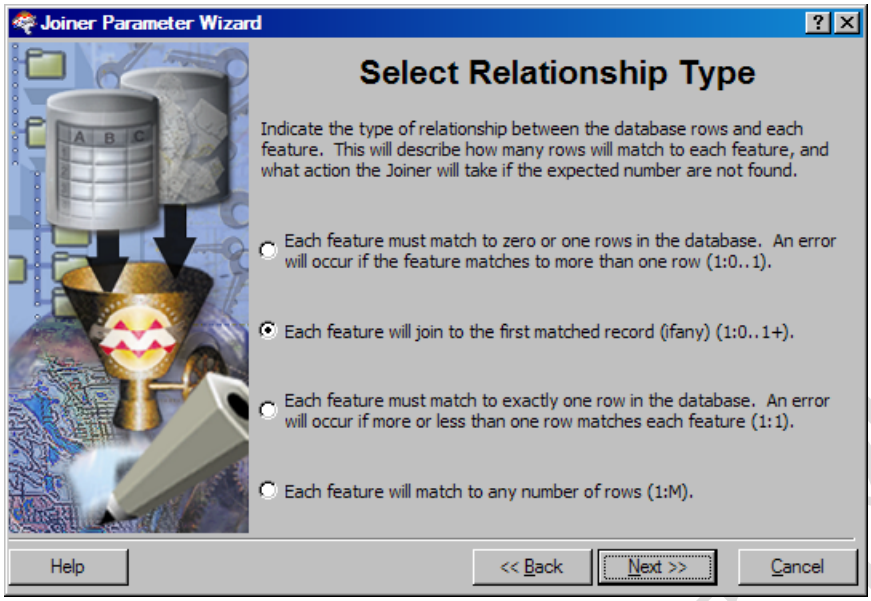
9、在下一个屏幕上，从每列上都选择type来创建一个关键对。



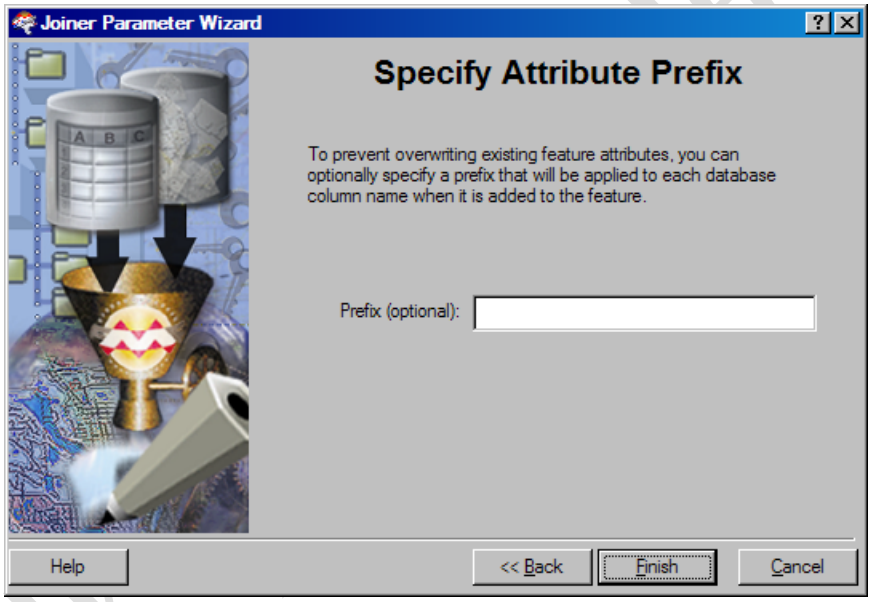
10、在下一个屏幕上，对这列选择igds_style作为添加到要素的属性。



11、在下一个屏幕上，选择连接第一个匹配记录作为关系。

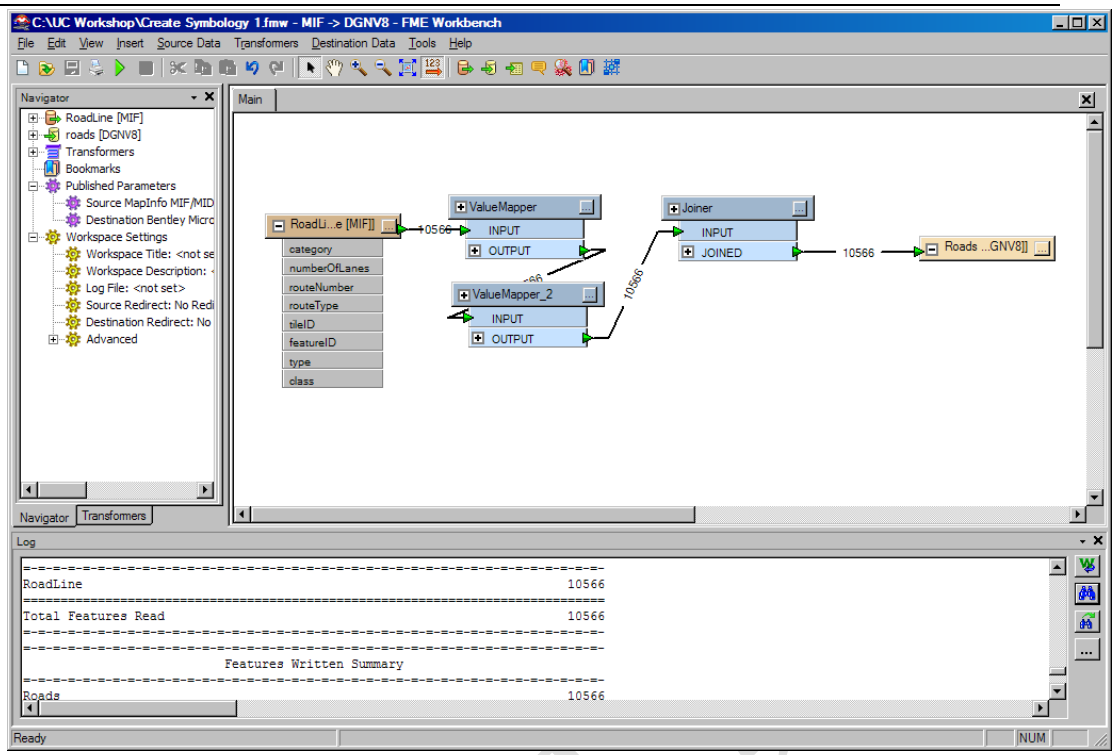


12、在下一个屏幕上，保留Prefix设置为空，点击Finish。

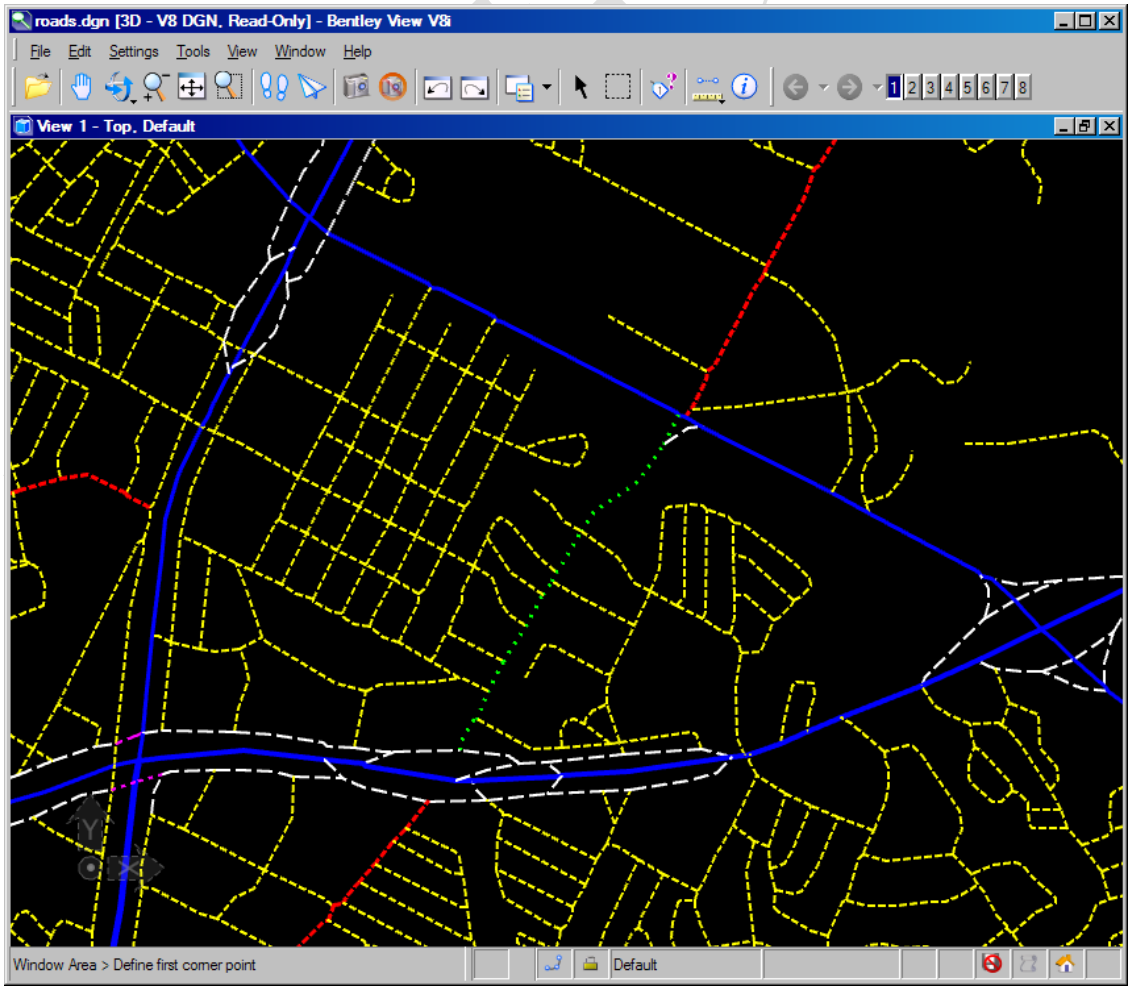


13、连接Joiner的输出端口到目标要素类Roads。

14、保存工作空间为Create Symbology.fmw，运行它。



15、在Bentley View中打开C:\FMEData\Output\workshop\roads.dgn，检验结果。



从符号中提取属性

保留前面在Workbench中打开的工作空间。打开另一个Workbench，创建一个新的工作空间。添加roads.dgn文件作为源MicroStation数据集，添加*C:\FMEData\Output\workshop\roads_sym.mdb*文件作为目标Geodatabase数据集。

绘制符号回属性，可以使用上一个练习中的相同的转换器完成。*ValueMappers*在属性中有一个设置可以反向映射，所以你只需改变源和目标的属性。你需要再次通过*Joiner*向导来翻转映射方向。保存工作空间为Extract Symbology.fmw，运行它。在FME Viewer中打开*C:\FMEData\Output\workshop\roads_sym.mdb*和*C:\FMEData\Data\Roads\RoadLine.mif*，对比下这两个数据集间的属性。

结束

这个教程包含了高级培训来完成GIS/CAD之间的转换，作为本次练习，使用你新学到的知识可以做到：

-- 在CAD中使用属性、标注和/或符号存储GIS属性。

-- 从CAD源文件中提取标注和/或符号来得到GIS属性。

也可以通过浏览Safe Software公司的FMEUserCentral网站来确认。这个网站也连接了其他的资源，这些资源可以在你对这一功能的学习中使用。

北京世纪安图