

実習 7 ジオメトリ間の空間的関係の判定・リスト属性

国土数値情報バスルートデータ（Esri Shapefile 形式ライン）、及び、これまでの実習と同じ市区町村区域データ（Esri Shapefile 形式ポリゴン）に基づき、路線バス（民間）の各バスルートラインに、そのラインが通過している「市区町村数」、「市区町村コード」、「市区町村名」を新たな属性として付加するためのワークスペースを作成します。

ひとつのバスルートラインが複数の市区町村を通過している場合、市区町村コード、市区町村名は、それぞれカンマ区切りで列挙することになります。

作成するワークスペースは、ひとつの都道府県の区域を対象とすることになります。以下、千葉県を例として説明しますが、他の都道府県について行っても構いません。

実行結果は FME Data Inspector で確認します。

はじめに、国土数値情報ダウンロードサービスサイト（JPGIS2.1）より千葉県のバスルートデータをダウンロード、解凍し、"N07-11_12.shp"（他の都道府県の場合はファイル名が異なります）が展開されたことを確認してください。

国土数値情報ダウンロードサービスサイト URL

<http://nlftp.mlit.go.jp/ksj/>

ソースデータ

N07-11_12.shp	Esri Shapefile 形式 バスルート（平成 23 年度千葉県）ライン
mmm20151001.shp	Esri Shapefile 形式 市区町村区域ポリゴン

国土数値情報ダウンロードサイトを参照して、バスルートデータの属性定義を確認しておきます。

国土数値情報バスルート Shapefile 形式データの属性

属性名	意味	説明
N07_001	バス区分	バス路線の運行形態による区分(1~5) 1:路線バス(民間), 2:路線バス(公営), 3:コミュニティバス, 4:デマンドバス, 5:その他
N07_002	事業者名	バス路線を運営する事業者名(コミュニティバスは自治体名)
N07_003	バス系統	バス路線の系統番号・系統名。系統が未整備であれば路線名・事業者名と連番。
N07_004	平日運行頻度	平日の一日当たりの運行本数の平均値(本/日)
N07_005	土曜日運行頻度	土曜日の一日の運行本数の平均値(本/日)
N07_006	日祝日運行頻度	日曜日、祝日の一日の運行本数の平均値(本/日)
N07_007	備考	バス路線に関する特記すべき事項

今回作成するワークスペースは次の 4 つの部分で構成されます。

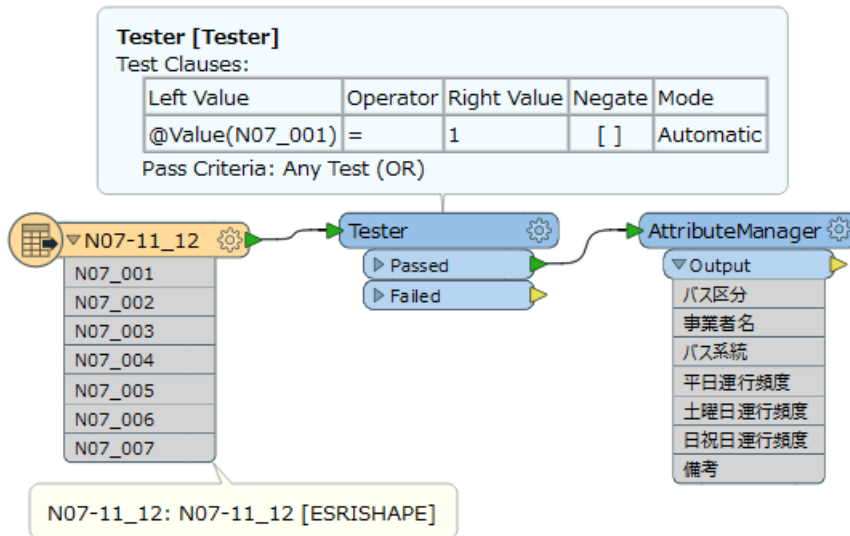
- 1) バスルートラインを読み込み、路線バス（民間）に該当するラインを抽出する。
- 2) 市区町村ポリゴンを読み込み、千葉県内の市区町村を抽出する。
- 3) 1)で抽出したバスルートラインと 2)で抽出した市区町村ポリゴンの空間的関係に基づき、各ラインに、それが通過している全ての市区町村の属性（市区町村コード、市区町村名）を与える。
- 4) 複数の市区町村コード、市区町村名をカンマ区切りで列挙した属性を作成する。

1)と 2)を行うためのデータフローはこれまでの実習で学んだ範囲で作成できるので、詳細な説明は省きます。今回の主題は 3)と 4)です。

7-1. 路線バス（民間）に該当するバスルートラインの抽出

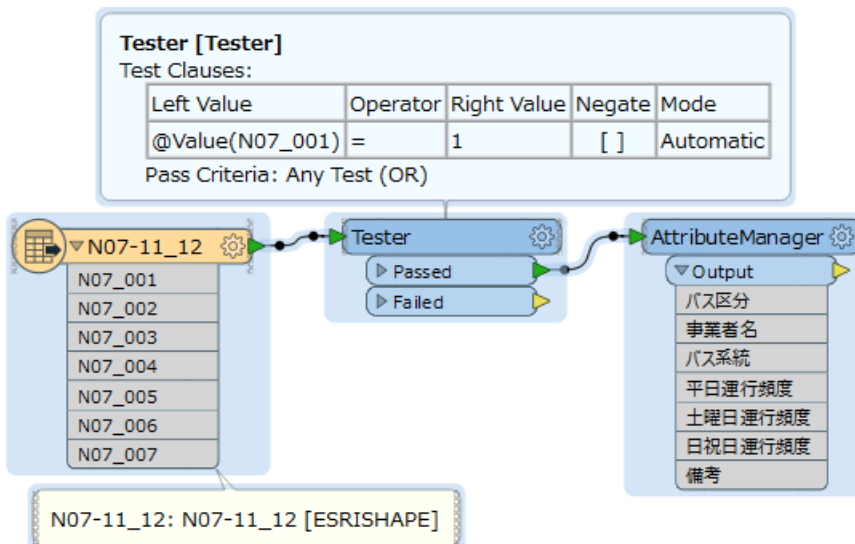
ソースデータ"N07-11_12.shp"（国土数値情報バスルート 千葉県）を読み込むためのリーダーとリーダーフィーチャータ입をワークスペースに追加し、**Tester** によってバス区分（N07_001）が 1 であるフィーチャー、つまり、路線バス（民間）に該当するラインのみを抽出します。

また、ソースデータの属性名は分かりにくいので、**AttributeManager** によって変更しておきます。

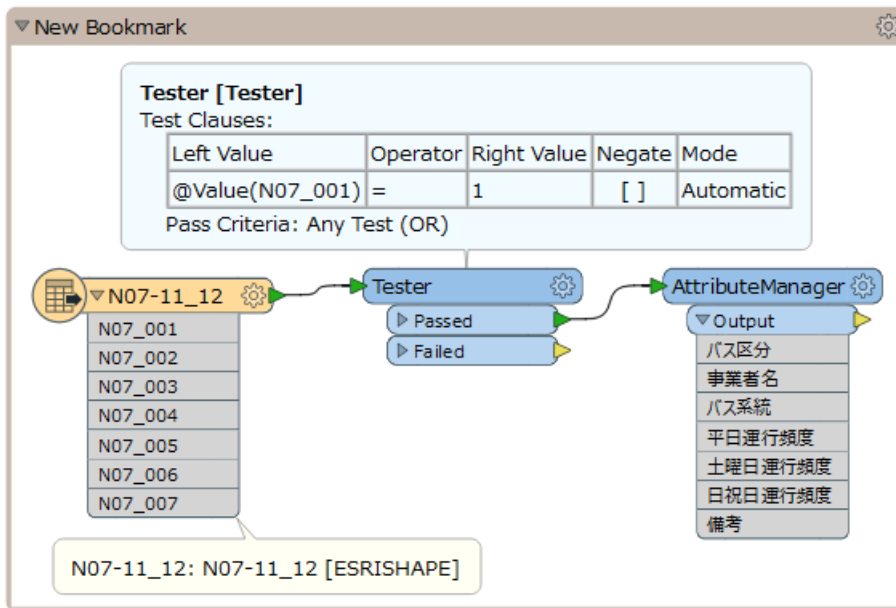


リーダーの追加やトランスフォーマーの接続、パラメーター設定に関して特筆すべきことはありませんが、ここで、**Bookmark**（ブックマーク）について説明します。

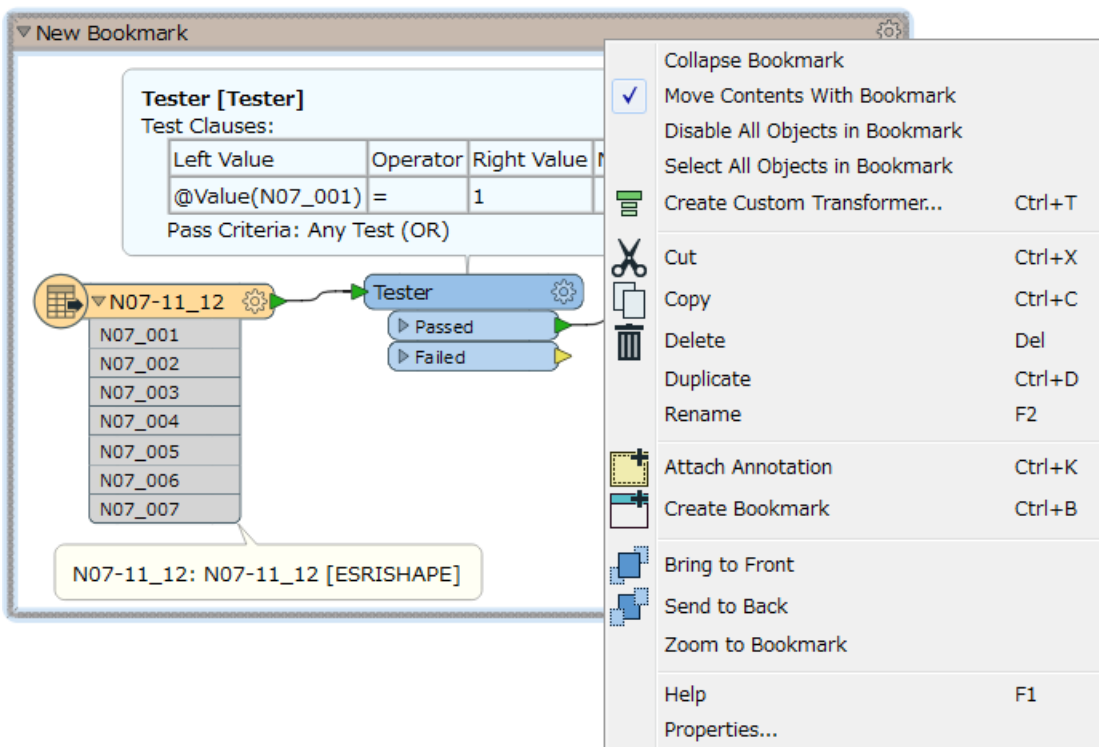
まず、ここまでに追加した全てのオブジェクトを選択してください。**Ctrl** キーを押しながらひとつひとつクリックすると複数のオブジェクトが選択できます。あるいは、選択したいオブジェクトを取り囲む矩形領域の対角線をマウスでドラッグすると、より素早く選択できます。



この状態で選択したオブジェクトのどれかひとつを右クリックして表示されるメニューで **Create Bookmark** を選択すると、次のように、選択したオブジェクトを取り囲む名前（タイトル）付きの枠が表示されます。これを **Bookmark**（ブックマーク）と呼んでいます。

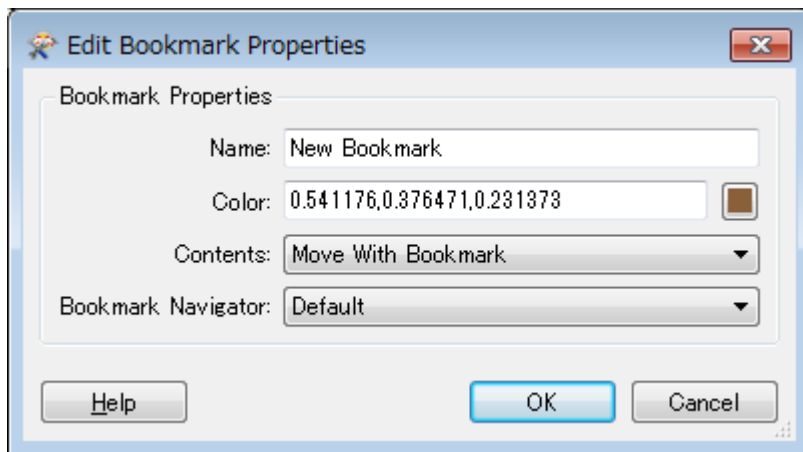


ブックマークを右クリックすると、そのブックマークを操作するためのメニューが表示されます。

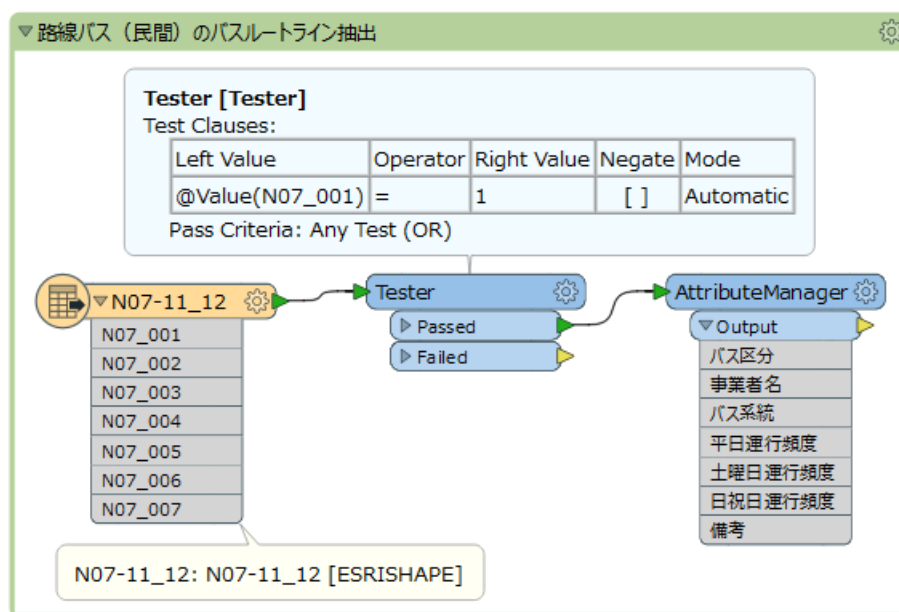


さしあたり、**Delete**（ブックマークの削除。内部のオブジェクトは削除されません）と **Properties**（プロパティ）を覚えてください。

このメニューで **Properties** を選択するか、または、ブックマーク右上のプロパティボタン（歯車アイコン）をクリックするとプロパティ編集画面が開き、ブックマークの名前（Name）と色（Color）が編集できます。



名前と色を変更してみましょう。



ブックマーク左上のアイコン（三角形）をクリックすると、ブックマークの折りたたみ/展開が交互に切り替わります。

ブックマークの有無によってワークスペースの処理内容に変わりはありませんが、キャンバス上でオブジェクトをグループ化することによって処理内容を分かり易く表現するために非常に効果的です。また、多数のプロセスで構成される大規模なワークスペースでは、プロセスごとにブックマークを作成し、編集以外のブックマークは折りたたんでおくことによって、ワークスペースの作成・保守がし易くなります。

Annotation（注釈）とあわせて活用してください。

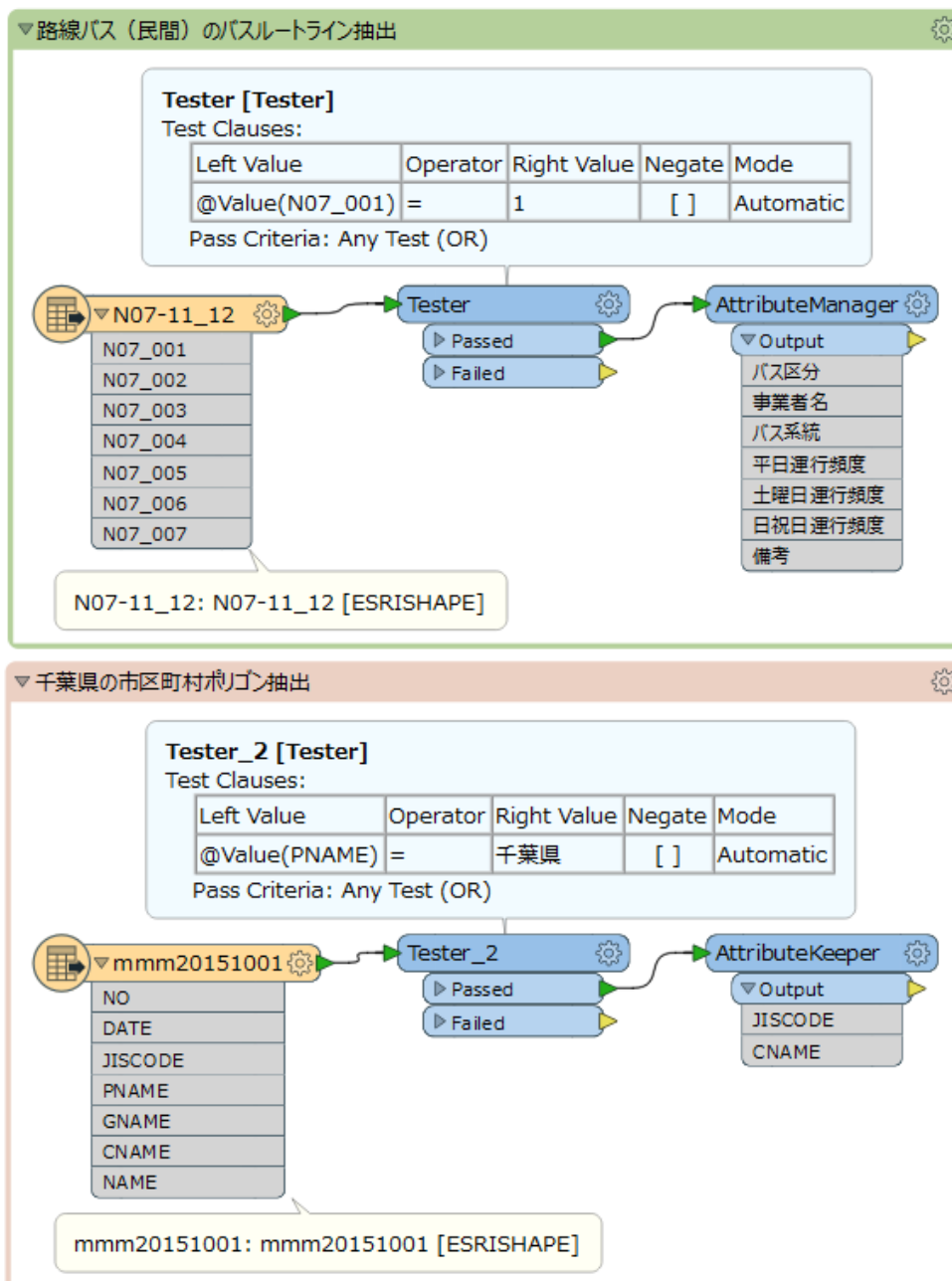
7-2. 千葉県内の市区町村ポリゴンの抽出

次に、ソースデータ"mmm20151001.shp" (市区町村ポリゴン) を読み込むためのリーダーとリーダーフィーチャータイプを追加し、Tester によって都道府県名 (PNAME) が「千葉県」であるポリゴンのみを抽出します。

また、AttributeKeeper によって、JISCODE (市区町村コード) と CNAME (市区町村名) 以外の属性を削除しておきます。

このデータフローもこれまでの実習で学んだ範囲で作成できるので、特筆すべきことはありません。ブックマークを追加し、先に作成したバスルートラインを抽出するためのデータフローの下に配置しましょう。

ここまでで次の 2 つのデータフローが作成されました。Inspector をそれぞれの末尾に接続、実行して、設計通りに働くことを確認してください。

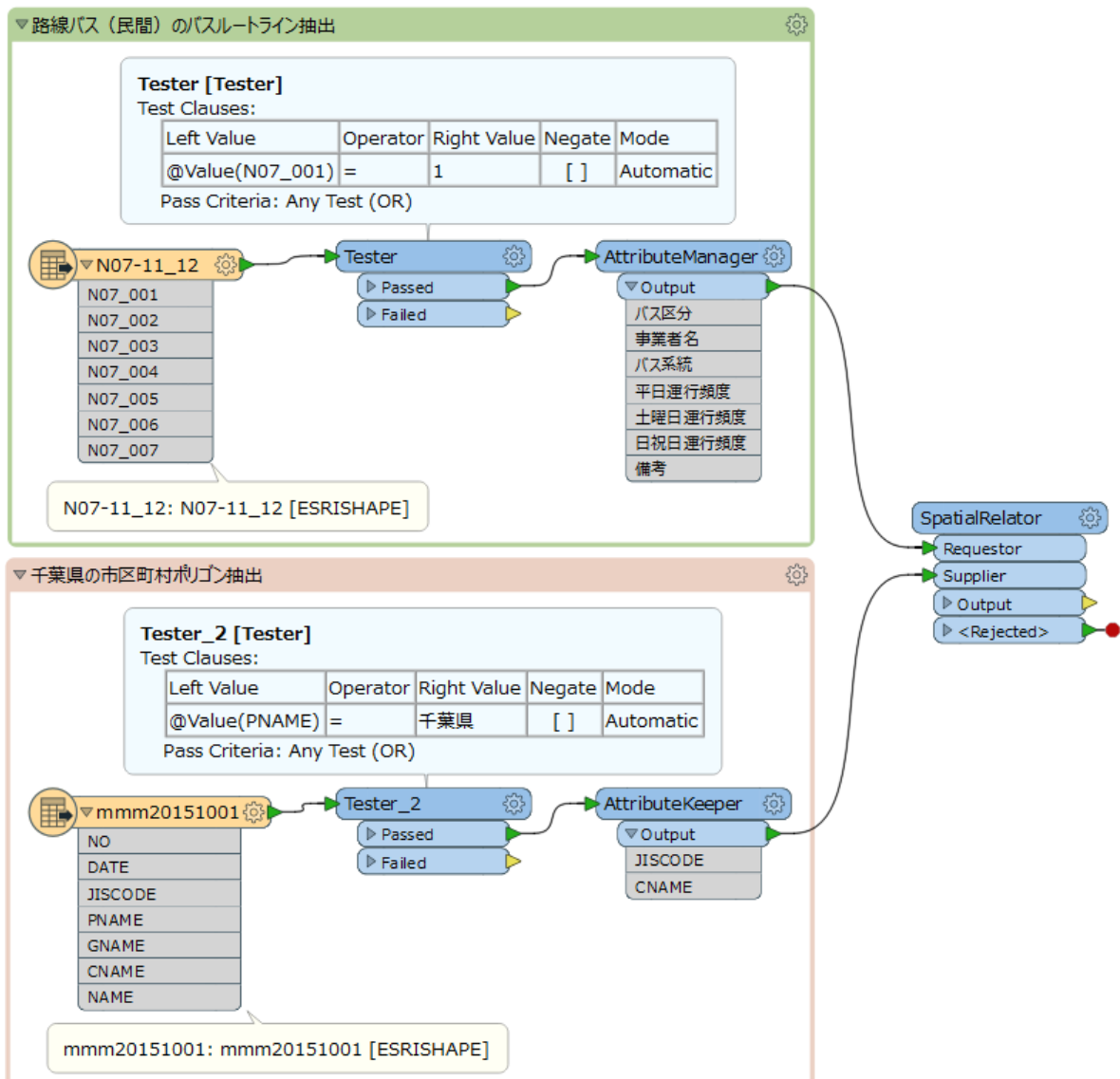


7-3. バスルートラインへの市区町村属性の結合

(1) SpatialRelator トランスフォーマーの追加と接続

各バスルートラインに、それが通過している全ての市区町村ポリゴンの属性を結合するには、SpatialRelator が使用できます。

SpatialRelator をワークスペースに追加し、Requestor ポートとバスルートラインのデータフロー末尾の AttributeRenamer、Supplier ポートと市区町村ポリゴンのデータフロー末尾の AttributeKeeper を接続してください。



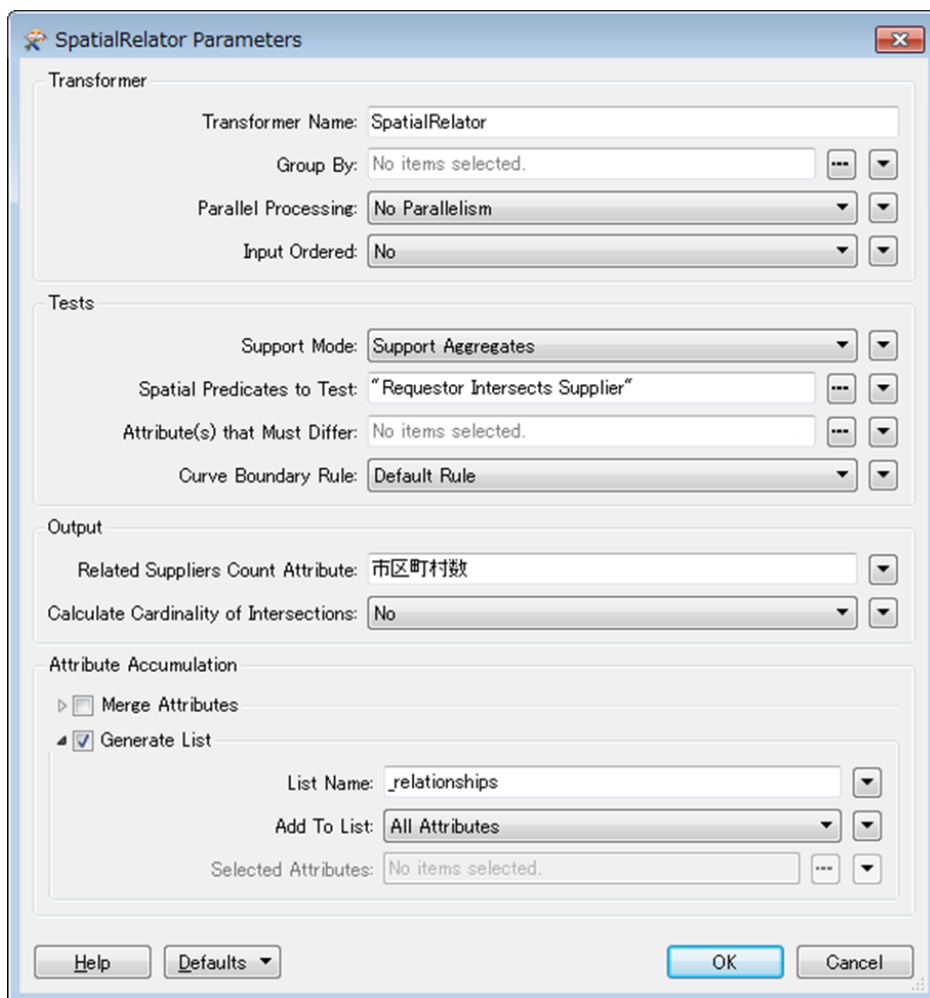
SpatialRelator のパラメーターは、次のように設定します。

Spatial Predicates to Test: "Requestor Intersects Supplier"

Related Suppliers Count Attribute: 市区町村数

Generate List (チェックボックスをオンにする)

List Name: _relationships



この設定により、SpatialRelator は、各々の Requestor フィーチャー (バスルートライン) について、Supplier フィーチャー (市区町村ポリゴン) との間で "Requestor Intersects Supplier" に該当する空間的関係の有無を調べ、その関係にあてはまる全ての Supplier フィーチャーの属性 (JISCODE と CNAME) を "_relationships" リストに格納して Requestor フィーチャーに結合します。

Output ポートからは、Supplier フィーチャーとの空間的関係の有無に関わらず、すべての Requestor フィーチャーが出力されます。

主要なパラメーターについて少し詳しく見ておきます。

1) Spatial Predicates to Test

検出したい空間的関係の種類を選択します。空間的関係の種類は次の表のように細かく分類できませんが、多くの場合、"Requestor Intersects Supplier" (Contains~Within のどれかひとつ以上に該当する) が適用できます。

関係	意味
Requestor Intersects Supplier	以下の関係の内いずれかひとつ以上に該当する。
Requestor Contains Supplier	Requestor が Supplier を含む。
Requestor Crosses Supplier *	Requestor と Supplier が交わる。
Requestor Equals Supplier	Requestor と Supplier が等しい。
Requestor Overlaps Supplier *	Requestor と Supplier が重なる (線と線の間、面と面の間)。
Requestor Touches Supplier	Requestor と Supplier が接する。
Requestor is Within Supplier	Requestor が Supplier に含まれる。

注: * 印の関係は、Support Mode パラメーターが "Support All Predicates" のときのみ選択できる。

2) Related Suppliers Count Attribute: 市区町村数

Requestor フィーチャー (バスルートライン) に対して空間的関係があった **Supplier** フィーチャー (市区町村ポリゴン) の数は、属性としてその **Requestor** フィーチャーに付加されます。**Related Suppliers Count Attribute** パラメーターには、その属性の名前を指定します。

3) List Name: _relationships

Generate List チェックボックスをオンにした場合、**Requestor** フィーチャー (バスルートライン) に対して空間的関係があった全ての **Supplier** フィーチャー (市区町村ポリゴン) の属性は、「リスト属性」(リスト属性については後で説明します) としてその **Requestor** フィーチャーに結合されます。**List Name** パラメーターには、そのリスト属性の名前を指定します。

ジオメトリ間の空間的関係の定義や判定に関しては、次のドキュメントも参照してください。

ジオメトリ間の空間的関係 (Spatial Relations)

<http://www.pragmatica.jp/fme/references/ReferenceSpatialRelations.html>

ジオメトリ間の空間的関係を判定する場合、それらの座標系 (測地系及び投影法) が同じであることを確認し、異なる場合は座標系を変換して統一する必要がありますが、国土数値情報ダウンロードサイトの説明より、国土数値情報バスルートデータは世界測地系 (JGD2000) 緯度経度で作成されており、市区町村ポリゴンと同じなので、そのまま比較することができます。

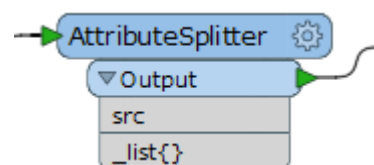
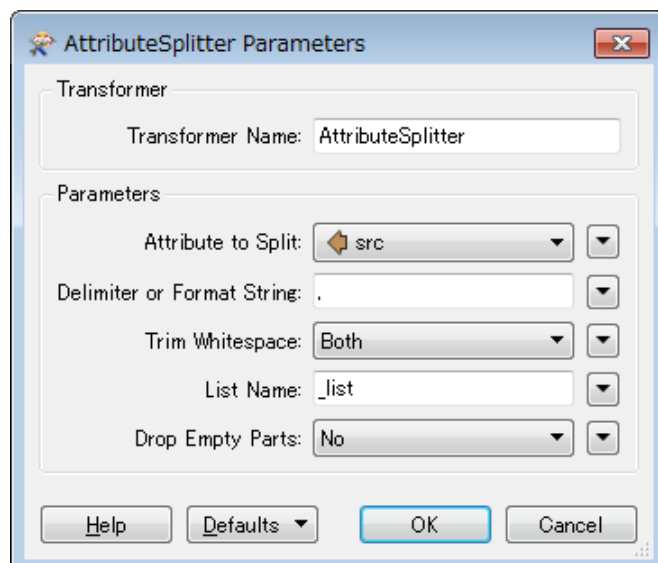
座標系の変換については実習 8 で取り上げます。

(2) リスト属性について

FME には、共通の名前 (リスト名) に { } で括ったインデクス (0 から始まる連番) を付加した属性名によって、関連のある複数の属性を一括して取り扱うための「リスト属性」(List Attribute) の仕組みがあります。

作成中のワークスペースでは使いませんが、**AttributeSplitter** は、処理結果をリスト属性に格納する代表的なトランスフォーマーのひとつです。これは、ある属性に格納されている文字列を指定した区切り文字 (列) で分割し、分割後の文字列を要素とするリスト属性を作成します。

例えば、パラメーターを次のように設定した場合、**src** 属性の値が "ABC,LMN,XYZ" であれば、"ABC", "LMN", "XYZ" を要素とするリスト属性 "_list{}" を作成します。



src 属性の値 = ABC,LMN,XYZ

作成されるリスト属性の要素

_list{0} = ABC

_list{1} = LMN

_list{2} = XYZ

リスト名とインデクスに加えて、ドットをはさんで連結したメンバー名によって、各要素を構成する個別の値を識別する構造のリストもあり、上記のような単純なリストと区別するときには「構造化リスト」(structured list または complex list) と呼ぶことがあります。構造化リストには、インデクスごとにグループ化された複数の属性を格納することができます。

リスト名{n}.メンバー1
 リスト名{n}.メンバー2
 リスト名{n}.メンバー3

SpatialRelator は、**Requestor** フィーチャー (バスルートライン) に、空間的関係のあった全ての **Supplier** フィーチャー (市区町村ポリゴン) の属性を構造化リストに格納して結合します。このとき、**List Name** パラメーターで指定した名前がリスト名になり、**Supplier** フィーチャーが持っていた属性の名前 (JISCODE, CNAME) がメンバー名になります。

例えば柏市 (市区町村コード=12217) と流山市 (市区町村コード=12220) を通っているバスルートラインに結合される市区町村名、市区町村コードは、リスト名が **_relationships** であるならば、次のような構造化リストに格納されます (インデクスの順番は異なるかも知れません)。

```
_relationships{0}.CNAME = 柏市
_relationships{0}.JISCODE = 12217
_relationships{1}.CNAME = 流山市
_relationships{1}.JISCODE = 12220
```

作成中のワークスペースに戻り、**SpatialRelator** の Output ポートの属性リストを展開すると、構造化リストが付加されていることが確認できます。

```
_relationships{}.JISCODE
_relationships{}.CNAME
```

また、**SpatialRelator** は、**Supplier** フィーチャーが持っていた属性だけでなく、検出された空間的関係に関する詳細な情報も、次のようなメンバー名で構造化リストに格納します。これらの内容についてはヘルプ等を参照してください。

```
_relationships{}.de9m
_relationships{}.pass{}
```

さらに、キャンパスのインターフェースには現れませんが、**Supplier** フィーチャーのフォーマット属性も構造化リストに格納されます。

```
_relationships{}.fme_basename
_relationships{}.fme_dataset
_relationships{}.fme_feature_type
```

など。

処理結果としてリスト属性を作成してフィーチャーに付加するトランスフォーマーは数多くあり、また、ワークスペースの目的を達成するためにリスト属性が重要な役割を担うケースはよくあります。次のページも参考にして、リスト属性の仕組みについてよく理解してください。

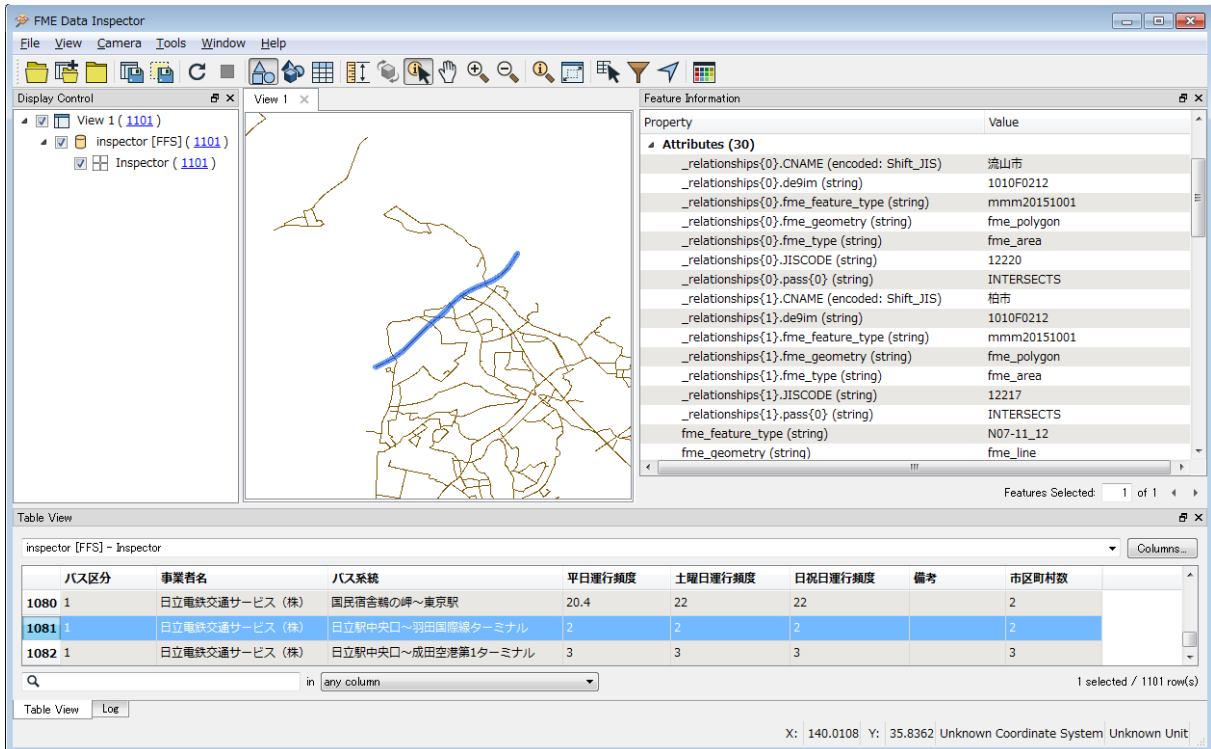
リスト属性 (List Attributes)

<http://www.pragmatica.jp/fme/references/BasicListAttributes.html>



SpatialRelator の Output ポートに Inspector を接続、実行してみましょう。

リスト属性の内容は Table View には表示されませんが、フィーチャーを選択したときに、Feature Information ウィンドウでそのフィーチャーが持っているリスト属性の内容が確認できます。



7-4. リスト属性の要素をカンマ区切りで列挙した属性の作成

SpatialRelator によって各バスルートラインに、それが通過する全ての市区町村ポリゴンの市区町村コード、市区町村名をリスト属性として結合できました。

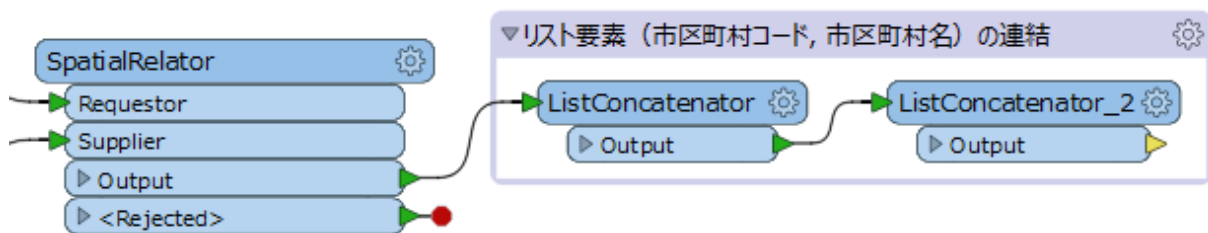
しかし、ワークスペースの処理結果をファイル等へ出力するときには、一般には、リスト属性に格納された各要素の値をそのまま書き出すことはできません。次のようなトランスフォーマーによって、リストではない属性（非リスト属性）に変換する必要があります。

リスト属性の要素を非リスト属性に変換する主なトランスフォーマー

トランスフォーマー	概要
ListConcatenator	リストの全要素を連結した文字列を格納した属性を作成する。
ListExploder	リストの要素数分フィーチャーをコピーし、コピー後のフィーチャーに各要素の値を格納した属性を与える。
ListIndexer	リスト内の特定のインデックスの要素を格納した属性を作成する。

ここでは **ListConcatenator** を使い、リスト属性の要素として格納されている複数の市区町村コード、市区町村名を、それぞれカンマ区切りで連結した文字列に変換することにします。

2 つの **ListConcatenator**（市区町村コード連結用と市区町村名連結用）を追加して、**SpatialRelator** の **Output** ポートに直列で接続してください。



1 つめの **ListConcatenator** では、次のパラメーター設定によって市区町村コードを連結します。

List Attribute: `_relationships{}.JISCODE`

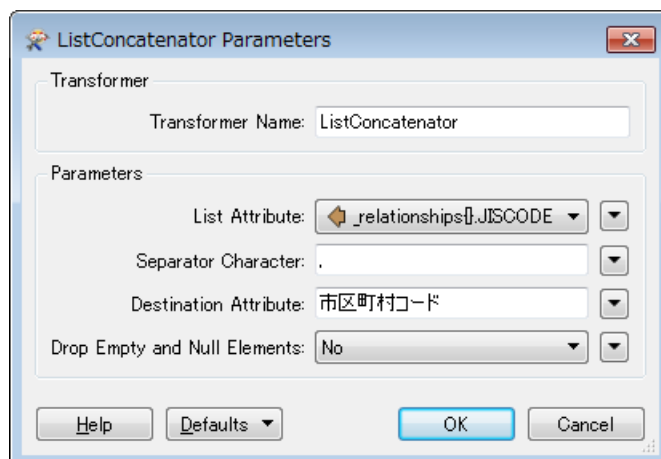
連結したい要素を格納しているリスト属性を選択します。構造化リストの場合は、メンバー名を含みます。

Separator Character: , (カンマ)

連結するときの要素間の区切り文字（列）を指定します。

Destination Attribute: 市区町村コード

連結した文字列を格納する属性名を指定します。

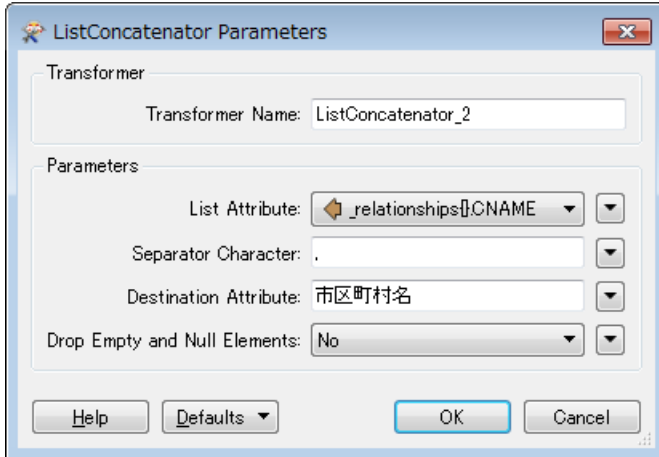


2つめの ListConcatenator で、同様に市区町村名を連結します。

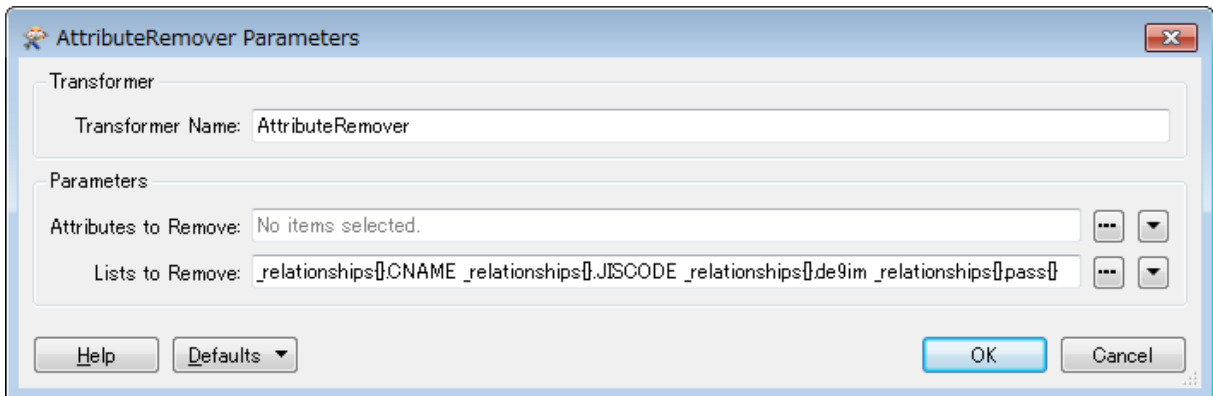
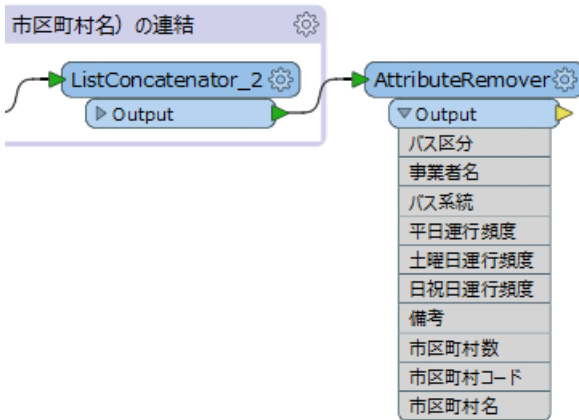
List Attribute: `_relationships{}.CNAME`

Separator Character: , (カンマ)

Destination Attribute: 市区町村名



必須ではありませんが、AttributeRemover で不要な属性（リスト属性を含む）を削除しておきましょう。AttributeRemover では、Lists to Remove パラメーターの設定（リスト属性の選択）によってリスト属性を削除することができます。



Inspector を追加して、AttributeRemover に接続すれば完成です。

実行後、FME Data Inspector の Table View 及び Feature Information で、複数の市区町村を通過しているラインでは、市区町村コード、市区町村名がカンマ区切りで列挙されていることを確認してください。

The screenshot shows the FME Data Inspector interface. The top part displays a map view with a red line representing a bus route. The bottom part shows the Table View, which contains a table of bus route data. The table has columns for route ID, bus division, company name, bus system, frequency, and municipality codes/names.

バス区分	事業者名	バス系統	平日運行頻度	土曜日運行頻度	日祝日運行頻度	備考	市区町村数	市区町村コード	市区町村名
1	(株) シティ...	立川駅~成田空港線	3	3	3		13	12409,12322,122...	芝山町, 酒々井...
2	(有) 大成交通	印旛日本橋大駅 京成臼井駅行	10	10	10		2	12231,12212	印西市, 佐倉市
3	(有) 大成交通	印旛日本橋大駅 大成交通前行	3	3	3		1	12231	印西市

実習 7 はここまでです。主に次の事項を学びました。

- ジオメトリ間の空間的関係の判定 SpatialRelator
- リスト属性
- リスト属性の要素の連結 ListConcatenator
- ブックマーク