FME Desktop®

Spatial Database Pathway Tutorial

FME 2013-SP2 Edition

SAFE SOFTWARE

This manual describes the functionality and use of the software at the time of publication. The software described herein, and the descriptions themselves, are subject to change without notice.

Copyright

Revisions

Every effort has been made to ensure the accuracy of this document. Safe Software Inc. regrets any errors and omissions that may occur and would appreciate being informed of any errors found. Safe Software Inc. will correct any such errors and omissions in a subsequent version, as feasible. Please contact us at:

Safe Software Inc.
Suite 2017, 7445 – 132nd Street
Surrey, BC
Canada
V3W1J8

www.safe.com

Safe Software Inc. assumes no responsibility for any errors in this document or their consequences, and reserves the right to make improvements and changes to this document without notice.

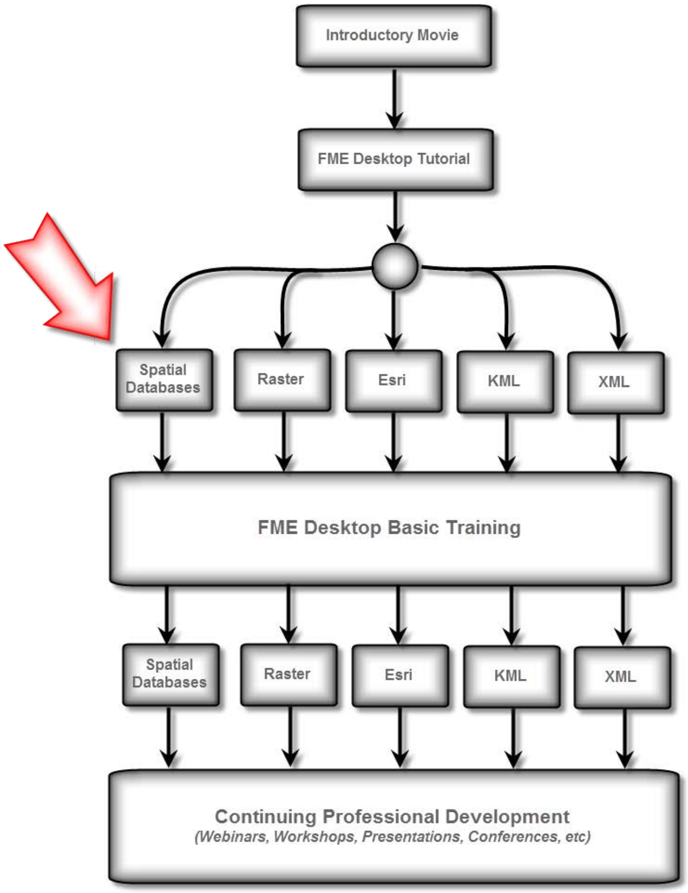
Trademarks

Documentation Information

Document Name: FME Desktop Database Pathway Tutorial
FME Version: FME 2013-SP2 (Build 13499) 32-bit
Operating System: Windows 7 SP-1, 64-bit
Database: Oracle Database Express 11g Release 2 (11.2), 32-bit
Database: SQL Server 2012
Database: PostGIS 2.0 (v2.0.1) for PostGreSQL x64 v9.2
Updated: July 2013

Introduction

Welcome to the FME Desktop Spatial Database Pathway Tutorial


Introductory Movie
FME Desktop Tutorial
Spatial Databases
Raster
Esri
KML
XML
FME Desktop Basic Training
Spatial Databases
Raster
Esri
KML
XML
Continuing Professional Development
(Webinars, Workshops, Presentations, Conferences, etc)

Spatial Database Pathway
This tutorial is an introduction to using spatial databases with FME. It is the first part of the FME Training Spatial Database Pathway.

It is assumed that you will already be familiar with the concepts and techniques described in the FME Desktop Tutorial.

***NB:** You can find the FME Desktop Tutorial online. It includes both PDF documents and a set of movies that cover each chapter.*

FME Version
This tutorial covers the use of FME Desktop® 2013 edition, specifically FME2013-SP2. Older versions of FME may not have some of the functionality described.

Sample Data
The sample data required to carry out the examples in this document can be obtained from:
http://www.safe.com/fmedata

Supported Databases
This tutorial was tested on, and includes documented steps for, the following databases:

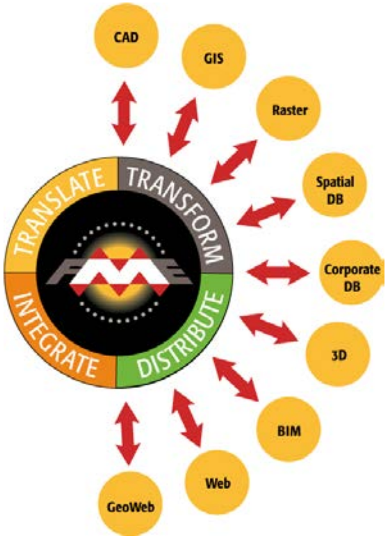
• PostGIS 2.0 for PostgreSQL 9.2	**FME Format**: PostGIS
• Oracle Express 11g (11.2)	**FME Format**: Oracle Spatial Object
• SQL Server 2012	**FME Format**: Microsoft SQL Server Spatial
• Enterprise Geodatabase	**FME Format**: Esri Geodatabase (ArcSDE Geodatabase)

For the purposes of simplicity, a PostGIS connection will be described and illustrated in all the following examples. Where other formats deviate strongly from the described steps, or where the step is specific to PostGIS, it will be highlighted with a tag:

PostGIS **Oracle** **Geodatabase** **SQL Server**

Introduction to FME

Here's a quick one-page reminder on what FME is.

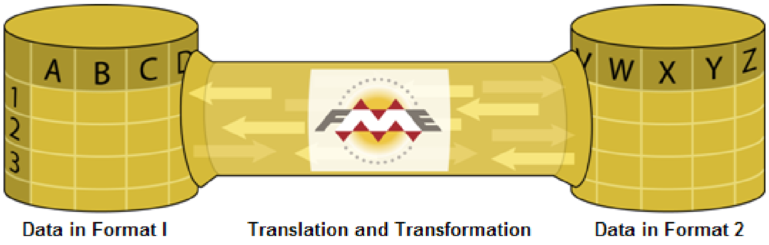


What is FME?
FME is a spatial data transformation platform that helps organizations more easily overcome a range of spatial data interoperability challenges. It is available in both desktop and server solutions.

FME is classified as a Spatial ETL (Extract-Transform-Load) tool, designed to help users master more spatial data transformation challenges than any other technology.

- *Extract* is the ability to read any format of spatial data.
- *Transform* is the ability to manipulate data during the translation process.
- *Load* is the ability to write the data in any other format.

With Data Transformation, the output from an FME process can be tailored to match a required structure, and can even be greater than the sum of the inputs.



The key FME Desktop application is **FME Workbench**, an intuitive point and click interface for graphically defining translations and transformations as a flow of data.

FME Quick Translator is an application for carrying out basic, non-customized translations.

FME Data Inspector is an application for visually inspecting both spatial and non-spatial data.

Spatial Database Basics

Here are some basic facts and information about Spatial Databases and their relationship with FME

Spatial Database Operations

Spatial databases are almost exclusively used for long-term data storage, rather than short-term data transfers, so the key operations are getting data into and out of that store.

The three key operations that occur with a spatial database are:

- Data Imports
- Data Updates
- Data Distribution

All of these actions involve not just transforming the data into the correct format, but also transforming data into the correct schema (data model).

Supported Database Formats

FME supports many database formats, including spatial database formats (vector and raster) and non-spatial database formats.

Some of the notable formats supported by FME include:

ESRI	ArcSDE, ArcSDE Raster, Geodatabase, Geodatabase Raster
Autodesk	FDO Providers
Google	Google Fusion Tables, Google Spreadsheet
IBM	DB2 Non-Spatial, DB2 Spatial, Informix, Informix Spatial
Intergraph	GeoMedia SQL Server Warehouse
MapInfo	SpatialWare
Microsoft	SQL Server Non-Spatial, SQL Server Spatial, Azure
Netezza	Netezza, Netezza Spatial
Oracle	Oracle Non-Spatial, Spatial Object, Spatial Point Cloud, Spatial GeoRaster
Postgres	PostgreSQL, PostGIS
Smallworld	Smallworld 3, Smallworld 4
Teradata	Teradata Non-Spatial, Teradata Spatial

Find the complete list of supported formats at www.safe.com/fme/format-search/.

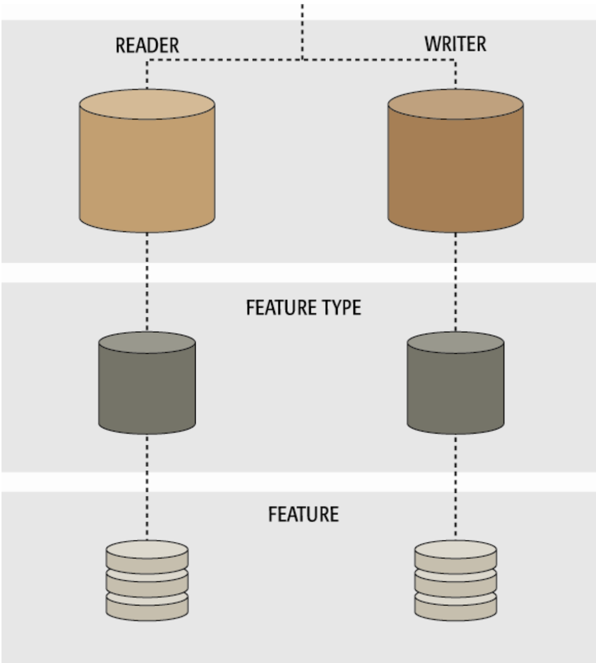
Important Database Terminology
It's important to clarify some of the basic database terms that are used by FME, as they may differ from what is used in a particular database package.

Schema
Also known as Data Model.

Coordinate System
Also known as Spatial Reference System or Spatial Reference Identifier (SRID).

Translation Components
An FME translation is made up of various components. When handling spatial databases, it is important to know what these components are, and to get a good grasp of the related FME terminology.

- **Readers** and **Writers** are the components that read and write data. They represent the database in a translation.
- **Feature Types** are the components that define data structures (or schemas). They represent the tables in a database.
- **Features** represent the individual records in a database table.



Connecting to a Spatial Database

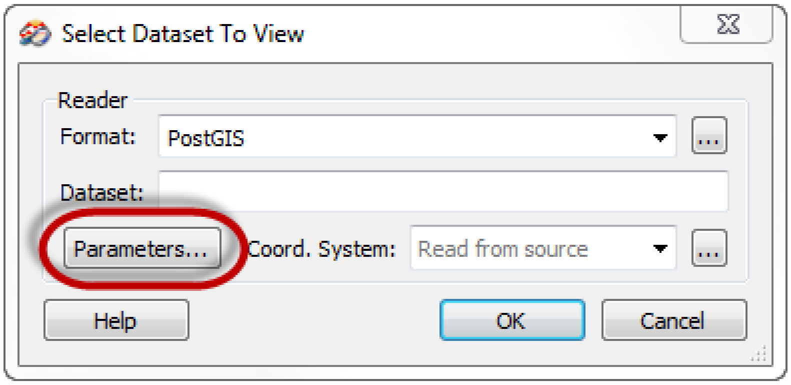
Database connections have slightly different parameters according to the format of data being used.

Connecting to a database is slightly different to selecting a file for a file/folder-based format. The operation relies much more on format specific parameters.

Basic Connection Parameters
The basic connection parameters are:

- Host (Server) Name
- Database (Service) Name
- Username
- Password
- Network Port Number

These parameters may differ slightly for each format, but will always be found in any dataset selection dialog by clicking on the “Parameters…” button.



Select Dataset To View

Reader

Format: PostGIS

Dataset:

Coord. System: Read from source

Troubleshooting
If a connection problem occurs, refer to the section titled *Troubleshooting*, at the end of this document.

PostGIS

Connecting to PostGIS

Connecting to a PostGIS database requires all five basic connection parameters.

Follow these steps to test the connection to your PostGIS database:

1) Start FME Data Inspector

Select **File > Open Dataset** from the menu bar to open the dataset selection dialog.

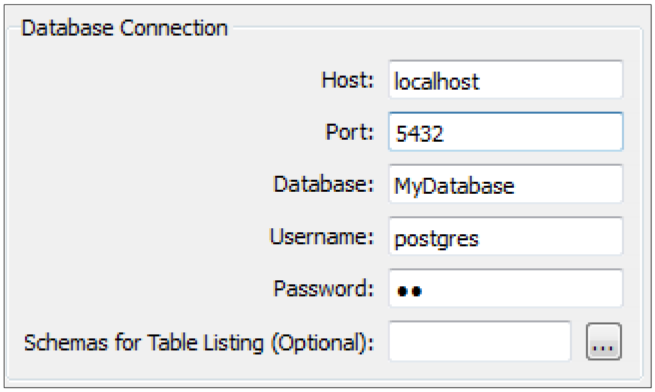
2) Define Dataset

When the dialog opens fill in the reader format field as follows:

Reader Format: PostGIS

Click the Parameters… button to open the parameters dialog.

Fill in the host, port, database, username, and password parameters.



Database Connection

Host: localhost

Port: 5432

Database: MyDatabase

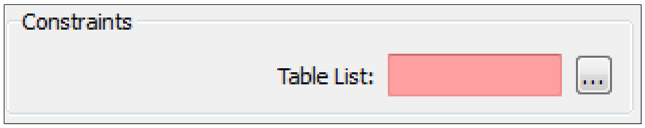
Username: postgres

Password: ••

Schemas for Table Listing (Optional):

3) Select Tables

Click on the browse button to the right of the Table List parameter.



Constraints

Table List:

- If the connection was successful, and there is already data in the database, then a list of tables will be presented. Select one and click OK (and then click OK on subsequent dialogs) to view the data.

- If the connection was successful, but there is no data in the database, then a dialog will open with a warning to this effect:

```
readSchema resulted in 0 schema features being returned
```

- If the connection was unsuccessful, then a dialog will appear with an error reporting the nature of the problem:

```
FATAL:  password authentication failed for user "postgres"
```

Oracle

Connecting to Oracle

Connecting to Oracle requires a client to be installed.
Connection is possible through either *tnsnames.ora* or a direct connection.

tnsnames.ora

tnsnames.ora is a file that usually resides in the Oracle client installation folder. It is a text file that consists of a number of service definitions of the form:

```
<net_service_name> =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <hostname>)(PORT = <1521>))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = <oracle_sid>)
    )
  )
```

Follow these steps to test the connection to your Oracle database using *tnsnames.ora*.

1) Start FME Data Inspector

Start the FME Data Inspector.
Select **File > Open Dataset** from the menu bar to open the dataset selection dialog.

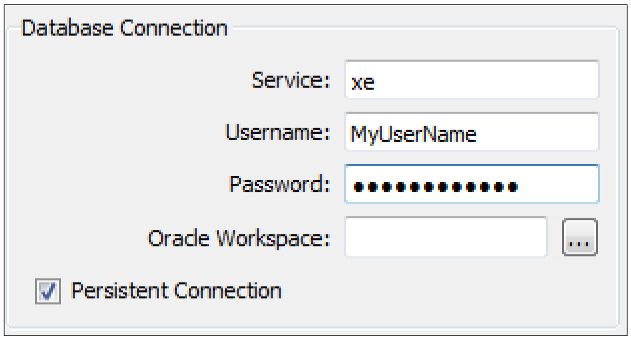
2) Define Dataset

When the dialog opens fill in the reader format field as follows:

Reader Format: Oracle Spatial Object

Click the Parameters… button to open the parameters dialog.

Fill in the service, username, and password.



Database Connection

Service: xe

Username: MyUserName

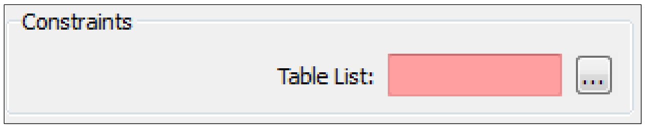
Password: ●●●●●●●●●●●●●

Oracle Workspace: ...

☑ Persistent Connection

3) Select Tables

Click on the browse button to the right of the Table List parameter.



Constraints

Table List: ...

- If the connection was successful, and there is already data in the database, then a list of tables will be presented. Select one and click OK to view the data.

- If the connection was unsuccessful, then a dialog will appear with an error reporting the nature of the problem:

```
Error connecting to Oracle database: message was
`ORA-12154: TNS:could not resolve the connect identifier specified'.
```

Oracle

Oracle Direct Connection

Direct Connection is when a single string is supplied that includes all the parameters required to connect to the database.

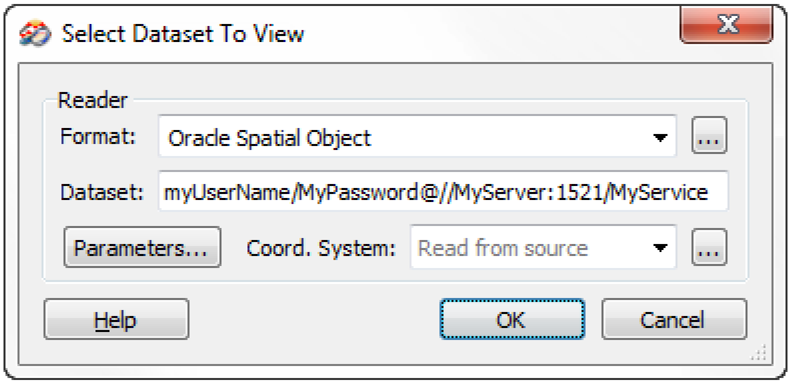
The connection string is of the form: `user/password@//hostname:port/sid`
For example: `training/training@//localhost:1521/xe`

Follow these steps to test the connection to your Oracle database using a direct connection.

1) Start FME Data Inspector
Start the FME Data Inspector.
Select **File > Open Dataset** from the menu bar to open the dataset selection dialog.

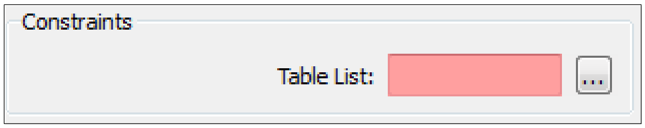
2) Define Dataset
When the dialog opens fill in the fields as follows:

Reader Format:	Oracle Spatial Object
Reader Dataset:	<username>/<password>@//<hostname>:<portnumber>/<service>


Select Dataset To View

Reader
Format: Oracle Spatial Object
Dataset: myUserName/MyPassword@//MyServer:1521/MyService
Coord. System: Read from source

3) Select Tables
Click the Parameters… button to open the parameters dialog.
Click on the browse button to the right of the Table List parameter.


Constraints
Table List:

- If the connection was successful, and there is already data in the database, then a list of tables will be presented. Select one and click OK to view the data.

- If the connection was unsuccessful, then a dialog will appear with an error reporting the nature of the problem; something along the lines of:

```
Error connecting to Oracle database: message was
`ORA-01017: invalid username/password; logon denied'.
```

SQL Server

Connecting to SQL Server

Connecting to a SQL Server database requires just four of the basic connection parameters. Port number is not required. There is an additional option to use Windows Authentication.

Connection Parameters

Follow these steps to test the connection to your SQL Server database using connection parameters:

1) Start FME Data Inspector

Start the FME Data Inspector.
Select **File > Open Dataset** from the menu bar to open the dataset selection dialog.

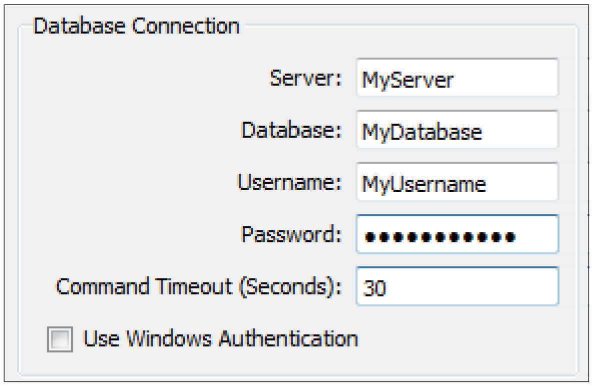
2) Define Dataset

When the dialog opens fill in the reader format field as follows:

Reader Format: Microsoft SQL Server Spatial

Click the Parameters… button to open the parameters dialog.

Fill in the server, database, username, and password parameters.



Database Connection

Server: MyServer

Database: MyDatabase

Username: MyUsername

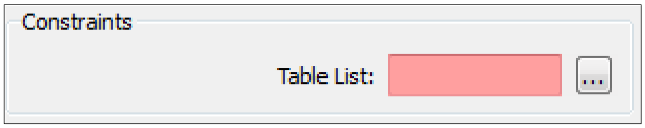
Password: ●●●●●●●●●●●

Command Timeout (Seconds): 30

□ Use Windows Authentication

3) Select Tables

Click on the browse button to the right of the Table List parameter.



Constraints

Table List:

- If the connection was successful, and there is already data in the database, then a list of tables will be presented. Select one and click OK to view the data.
- If the connection was unsuccessful, then a dialog will appear with an error reporting the nature of the problem:

```
MS SQL Server (Spatial) Reader: Connection failed.
Connection string `Provider=SQLOLEDB;Data Source=maul;Initial
Catalog=support; User ID=training;Password=training'.
Provider error `Login failed for user 'training'.'
```

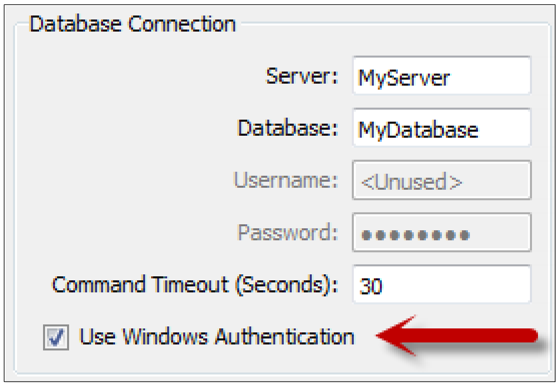
SQL Server

Windows Authentication

Connecting to a SQL Server database using Windows authentication is a similar process to connecting through parameters.

The main difference is that when the Windows Authentication option is selected, the Username and Password parameters are grayed-out.

Username and password will get supplied to the SQL Server database from the user's Windows login information.



Database Connection

Server: MyServer

Database: MyDatabase

Username: <Unused>

Password: ●●●●●●●●

Command Timeout (Seconds): 30

☑ Use Windows Authentication

Geodatabase

Connecting to an Enterprise Geodatabase

Connecting to an Esri Enterprise Geodatabase can be done either through a set of parameters, through OS Authentication, or through a Connection File.

Connection Parameters

Follow these steps to test the connection to your Geodatabase using connection parameters:

1) Start FME Data Inspector

Start the FME Data Inspector.
Select **File > Open Dataset** from the menu bar to open the dataset selection dialog.

2) Define Dataset

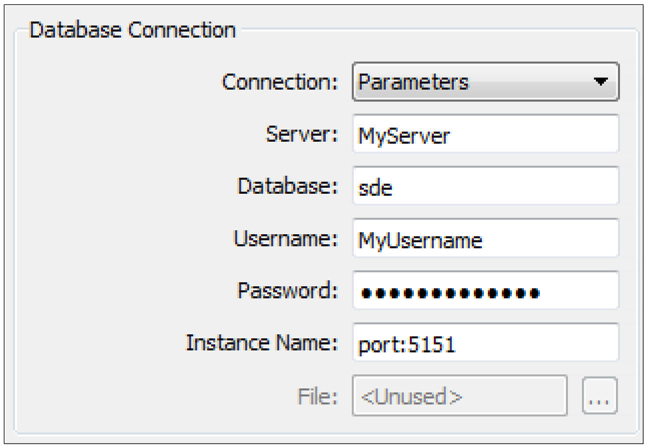
When the dialog opens fill in the reader format field as follows:

Reader Format: Esri Geodatabase (ArcSDE Geodatabase)

Click the Parameters… button to open the parameters dialog.

Ensure 'parameters' is selected in the connection type drop down list.

Fill in the server, database, username, and password parameters, plus the instance name.



Database Connection

Connection: Parameters

Server: MyServer

Database: sde

Username: MyUsername

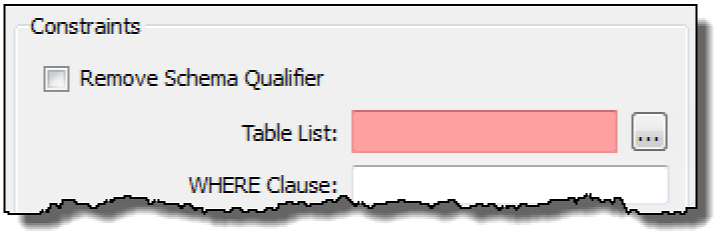
Password: ●●●●●●●●●●●●●

Instance Name: port:5151

File: <Unused>

3) Select Tables

Click on the browse button to the right of the Table List parameter.



Constraints

□ Remove Schema Qualifier

Table List:

WHERE Clause:

- If the connection was successful, and there is already data in the database, then a list of tables will be presented. Select one and click OK to view the data.
- If the connection was unsuccessful, then a dialog will appear with an error reporting the nature of the problem:

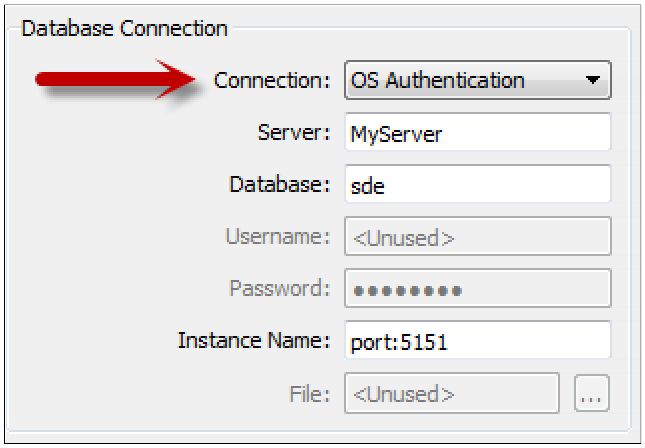
```
Could not open the Enterprise Geodatabase.
Please check that the connection parameters specified are correct.
The error number from ArcObjects is: '-2147216118'.
The error message from ArcObjects is: {Bad login user}
```

OS Authentication

Connecting to an Esri Geodatabase using Operating System authentication is a similar process to connecting through parameters.

The main difference is that when the OS Authentication option is selected, the Username and Password parameters are grayed-out.

Username and password will get supplied to the SQL Server database from the user's OS login information.



Database Connection

Connection: OS Authentication

Server: MyServer

Database: sde

Username: <Unused>

Password: ●●●●●●●●

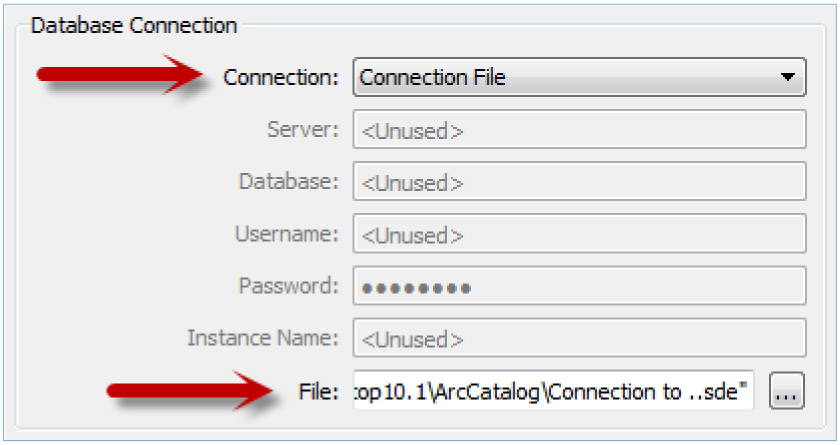
Instance Name: port:5151

File: <Unused>

Connection File

Connection files are a store of connection information. They are created by ArcGIS when a connection is made in one of the ArcGIS applications, and can be used by FME to connect.

The files are usually stored in the ESRI ArcCatalog user application directory; for example:
<User_Directory>\AppData\Roaming\ESRI\Desktop10.1\ArcCatalog\



Database Connection

Connection: Connection File

Server: <Unused>

Database: <Unused>

Username: <Unused>

Password: ●●●●●●●●

Instance Name: <Unused>

File: :op10.1\ArcCatalog\Connection to ..sde"

Select the Connection File method and then use the file browser to select the connection file to be used.

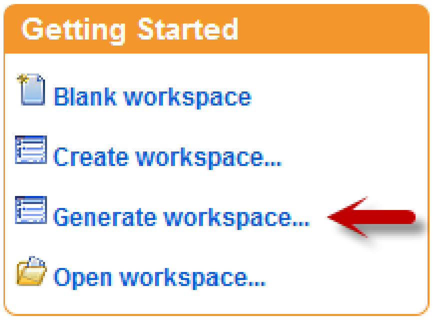
Writing to a Spatial Database

Simply writing data into a new table in a database is not much more complex than writing to a file format.

Writing Data

Writing to a database – rather than reading – is perhaps the most important topic for a new FME user, because data can't be read from a database until it has been written!

Follow these steps to load some parks data into your database:



Getting Started

- Blank workspace
- Create workspace...
- Generate workspace...
- Open workspace...

1) Start FME Workbench.

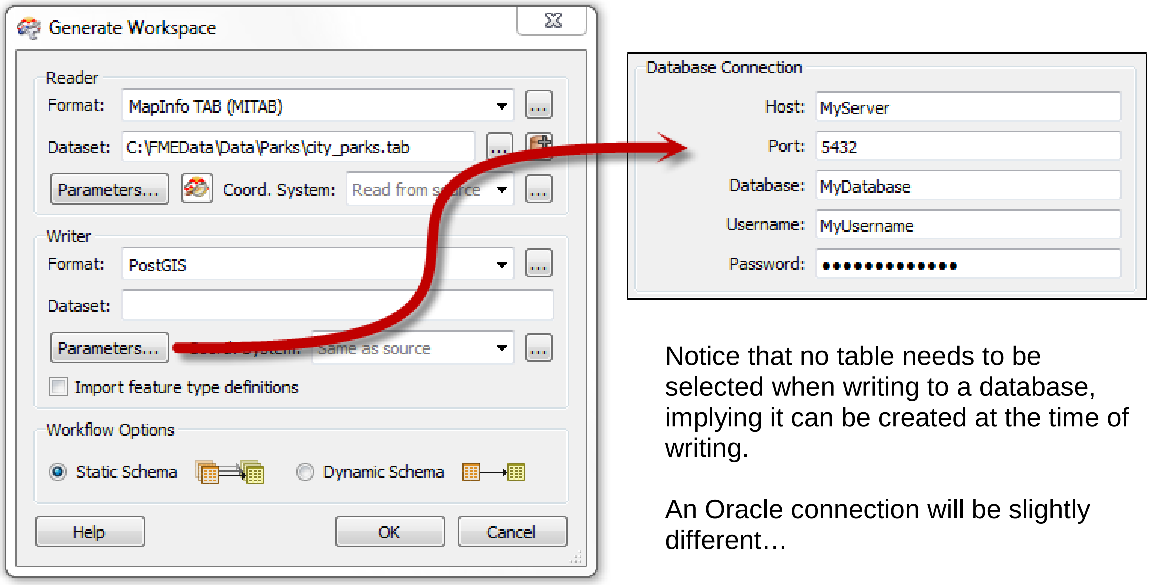
Start FME Workbench. In the startup tab choose the option to 'Generate workspace…'

When the Generate Workspace dialog opens, fill in the fields as follows:

Reader Format: MapInfo TAB (MITAB)
Reader Dataset: *C:\FMEData\Data\Parks\city_parks.tab*

Writer Format: <database format of choice>

For the writer format, choose your own database type, click the parameters button and then enter the connection parameters as tested earlier. Again, for the purposes of simplicity, a PostGIS connection will be illustrated:



Generate Workspace

Reader
Format: MapInfo TAB (MITAB)
Dataset: C:\FMEData\Data\Parks\city_parks.tab
Coord. System: Read from source

Writer
Format: PostGIS
Dataset:
Coord. System: Same as source
□ Import feature type definitions

Workflow Options
● Static Schema ○ Dynamic Schema

Database Connection
Host: MyServer
Port: 5432
Database: MyDatabase
Username: MyUsername
Password: •••••••••••••

Notice that no table needs to be selected when writing to a database, implying it can be created at the time of writing.

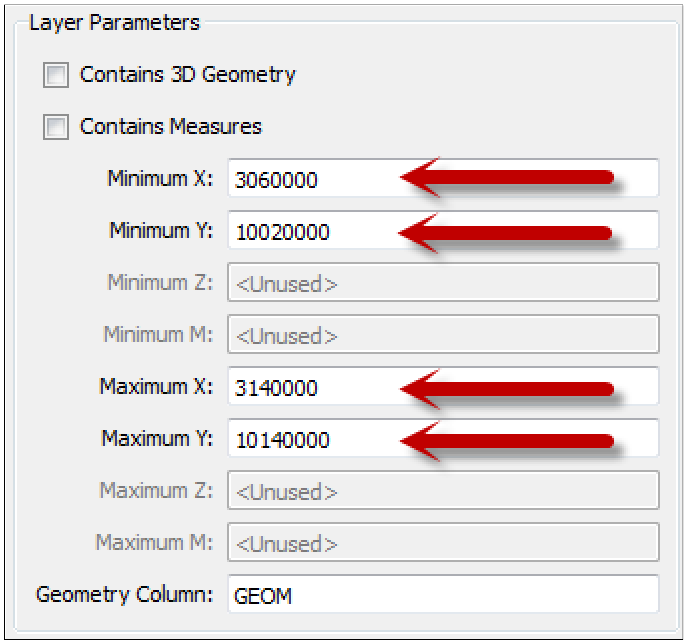
An Oracle connection will be slightly different…

Oracle

With Oracle, it is necessary to define the geographic extent to be covered by the table.

The default values are latitude and longitude, and will not work for this example. Therefore the minimum and maximum spatial extents must be defined.

Minimum X	3060000
Minimum Y	10020000
Maximum X	3140000
Maximum Y	10140000



Layer Parameters

□ Contains 3D Geometry

□ Contains Measures

Minimum X: 3060000

Minimum Y: 10020000

Minimum Z: <Unused>

Minimum M: <Unused>

Maximum X: 3140000

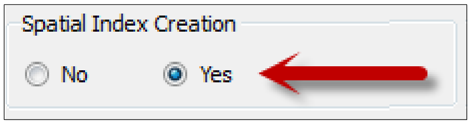
Maximum Y: 10140000

Maximum Z: <Unused>

Maximum M: <Unused>

Geometry Column: GEOM

Also, to enable FME to read and clip data at a later stage, it's necessary to create a spatial index; so set Spatial Index Creation = Yes



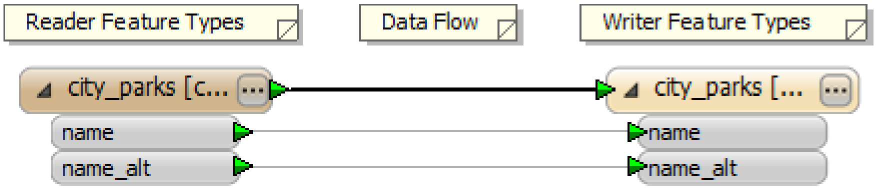
Spatial Index Creation

○ No ● Yes

All other parameters can retain their default values.

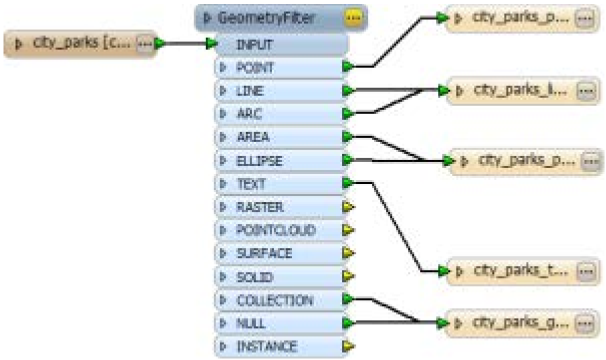
2) Create Workspace
Click OK to accept the selection and create a workspace.

The workspace will look something like this:



Geodatabase

For Geodatabase, the workspace will look notably different:



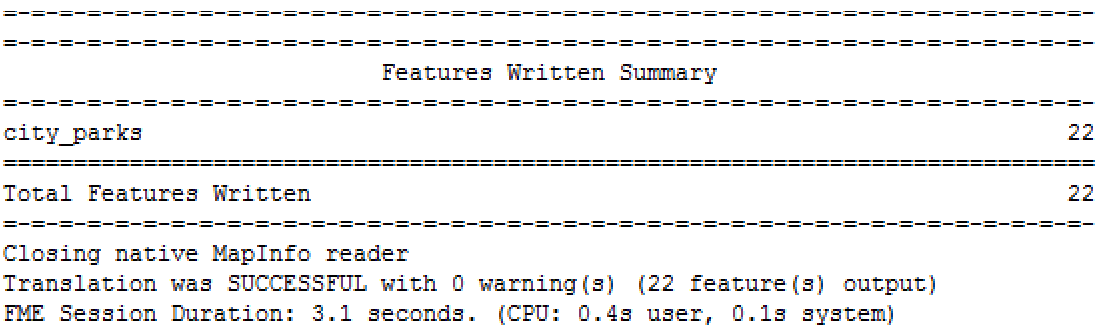
That's because only one type of geometry (point, line, polygon) can get written to any one Geodatabase table.

FME ensures this requirement by filtering the data and writing each geometry type to separate tables.

3) Save and Run Workspace

Save and then run the workspace.

The process should take just a few seconds, and load 22 records into the database.



```
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
                           Features Written Summary
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
city_parks                                                                  22
==============================================================================
Total Features Written                                                      22
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Closing native MapInfo reader
Translation was SUCCESSFUL with 0 warning(s) (22 feature(s) output)
FME Session Duration: 3.1 seconds. (CPU: 0.4s user, 0.1s system)
```

4) Inspect Output

Start the FME Data Inspector. Select **File > Open Dataset** from the menubar.

Define the format and dataset using the same technique for your database as in the section 'Connecting to a Spatial Database'.

This time, when selecting a table from the list, select the newly created city_parks table.

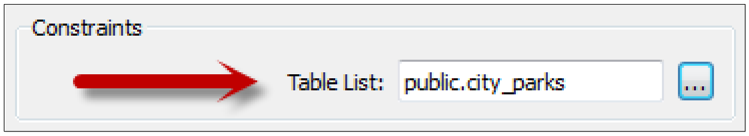
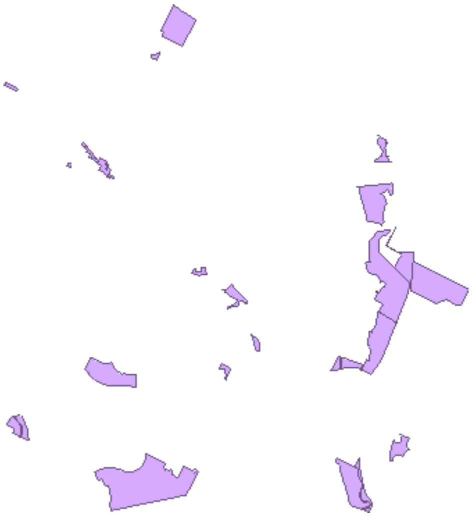


Table List: public.city_parks

Geodatabase

Because a different table is created for each geometry, the table name in Geodatabase will be city_parks_polygon

The spatial data will look something like this:



The FME Data Inspector is used for *Data Inspection*. This is the act of viewing data for verification purposes, before, during or after a translation.

Inspection can be used to verify:

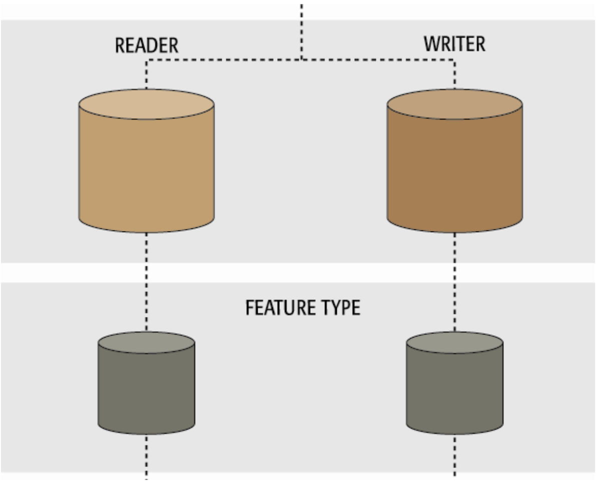
- Data Quantity
- Data Quality
- Schema (Tables and Attributes)
- Coordinate System

Controlling Database Writing

Various FME parameters exist to provide fine control when writing to a spatial database.

Just as there are different components to an FME translation, each component has its own set of parameters to control it.

So – when writing data – there are parameters to control the Writer (how FME handles the database as a whole) and also parameters to control the Feature Types (how FME handles individual tables).



Remember!

Reader/Writer = Database
Feature Type = Table

Writer Parameters

Writers are represented in the Navigator window of Workbench (by default the left-hand window) by a small, yellow-orange icon:

Clicking the little expand icon next to the Writer reveals all the parameters associated with it.

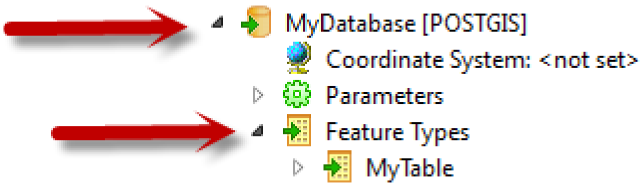
Each different format of Writer has its own set of parameters, many of which control connection parameters for the database.

When a format of data allows you to write data to either a Geography or Geometry column, you'll find a Writer parameter to control that in the Navigator window.

Feature Type Parameters

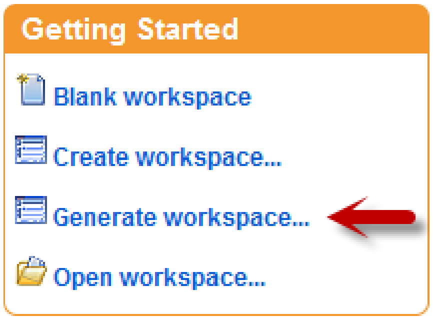
Feature Types are represented in the Navigator window of Workbench by a slightly different icon:

A Feature Type is revealed by expanding the Reader or Writer to which it belongs.



MyDatabase [POSTGIS]
Coordinate System: <not set>
Parameters
Feature Types
MyTable

Expanding the Feature Type reveals all the parameters associated with it. Again, each different format has its own set of Feature Type parameters.

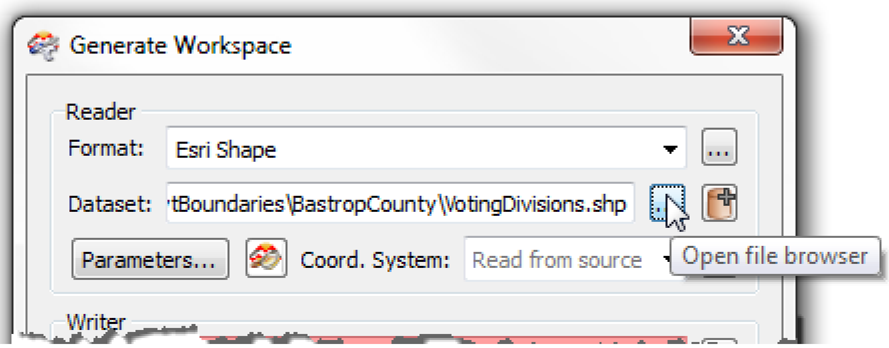


Follow these steps to practice using Writer and Feature Type parameters:

1) Start FME Workbench.
Start FME Workbench. In the startup tab choose the option to ‘Generate workspace…’

When the Generate Workspace dialog opens, fill in the fields as follows:

Reader Format:	ESRI Shape
Reader Dataset:	*C:\FMEData\Data\GovtBoundaries\BastropCounty\CensusTracts.shp* *C:\FMEData\Data\GovtBoundaries\BastropCounty\CountySubdivisions.shp* *C:\FMEData\Data\GovtBoundaries\BastropCounty\VotingDivisions.shp*
Writer Format:	<database format of choice>


Generate Workspace

Reader

Format: Esri Shape

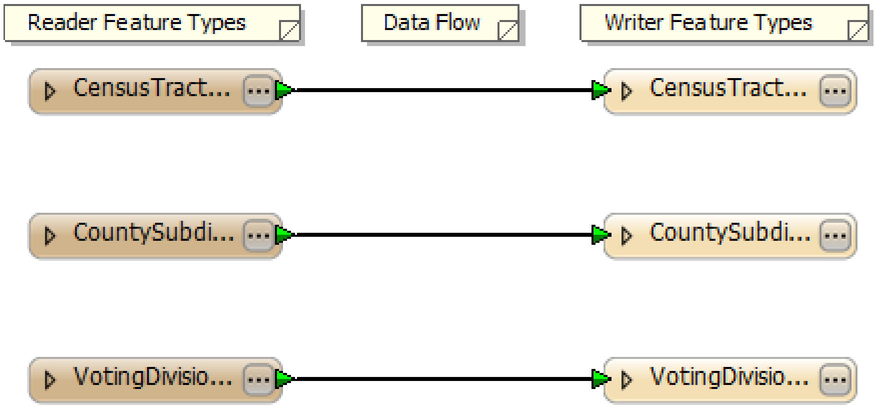
Dataset: ·tBoundaries\BastropCounty\VotingDivisions.shp

Coord. System: Read from source

For the Reader dataset, click the File Browser button, browse to the correct folder, and then select all three Shape files in the folder.

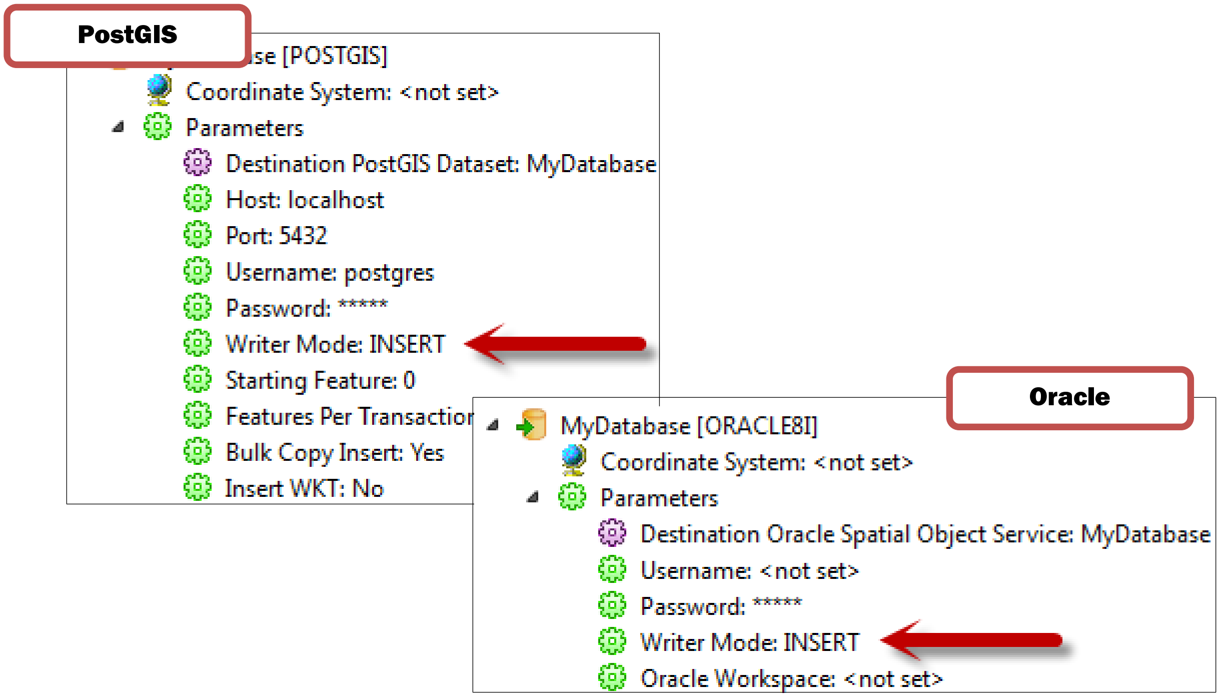
For the Writer format, choose your own database type, click the parameters button and then enter the connection parameters as tested earlier.

Click **OK** to create the workspace. Its layout will have three Feature Types on the Reader, to represent the incoming Shape files, and three Feature Types on the Writer to represent the database tables to be written.



2) Set Writer Parameters

Locate the database writer in the Workbench Navigator window.
Expand the list of parameters, and locate the Writer Mode parameter. Its position will vary according to the database format being used:

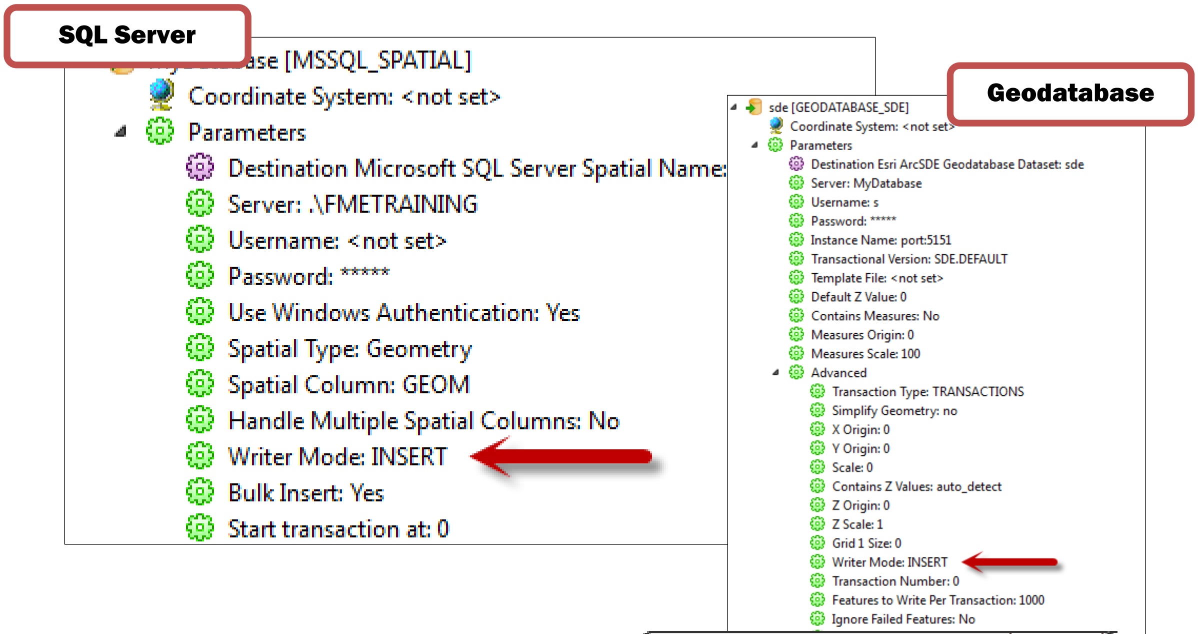


PostGIS

```
se [POSTGIS]
    Coordinate System: <not set>
    Parameters
        Destination PostGIS Dataset: MyDatabase
        Host: localhost
        Port: 5432
        Username: postgres
        Password: *****
        Writer Mode: INSERT
        Starting Feature: 0
        Features Per Transactio
        Bulk Copy Insert: Yes
        Insert WKT: No
```

Oracle

```
MyDatabase [ORACLE8I]
    Coordinate System: <not set>
    Parameters
        Destination Oracle Spatial Object Service: MyDatabase
        Username: <not set>
        Password: *****
        Writer Mode: INSERT
        Oracle Workspace: <not set>
```



SQL Server

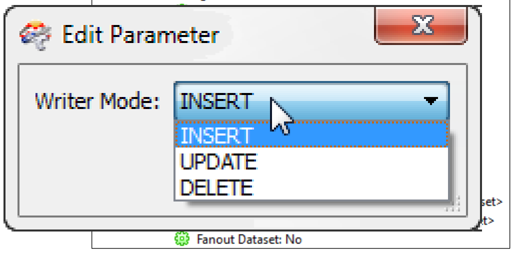
```
ase [MSSQL_SPATIAL]
    Coordinate System: <not set>
    Parameters
        Destination Microsoft SQL Server Spatial Name:
        Server: .\FMETRAINING
        Username: <not set>
        Password: *****
        Use Windows Authentication: Yes
        Spatial Type: Geometry
        Spatial Column: GEOM
        Handle Multiple Spatial Columns: No
        Writer Mode: INSERT
        Bulk Insert: Yes
        Start transaction at: 0
```

Geodatabase

```
sde [GEODATABASE_SDE]
    Coordinate System: <not set>
    Parameters
        Destination Esri ArcSDE Geodatabase Dataset: sde
        Server: MyDatabase
        Username: s
        Password: *****
        Instance Name: port:5151
        Transactional Version: SDE.DEFAULT
        Template File: <not set>
        Default Z Value: 0
        Contains Measures: No
        Measures Origin: 0
        Measures Scale: 100
        Advanced
            Transaction Type: TRANSACTIONS
            Simplify Geometry: no
            X Origin: 0
            Y Origin: 0
            Scale: 0
            Contains Z Values: auto_detect
            Z Origin: 0
            Z Scale: 1
            Grid 1 Size: 0
            Writer Mode: INSERT
            Transaction Number: 0
            Features to Write Per Transaction: 1000
            Ignore Failed Features: No
            Fanout Dataset: No
```

Double-click the Writer Mode parameter and it will open up a dialog in which to set it.

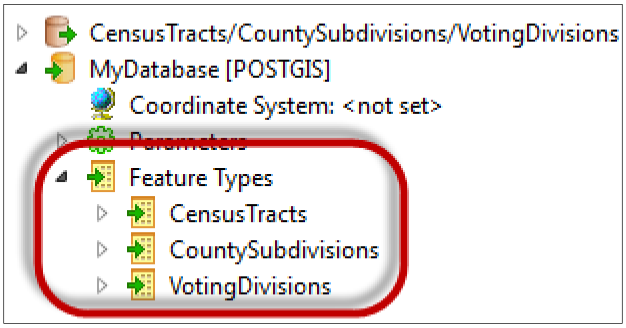
Make sure the mode is set to INSERT (because the aim here is to insert data), but also note there are options for DELETE and UPDATE.



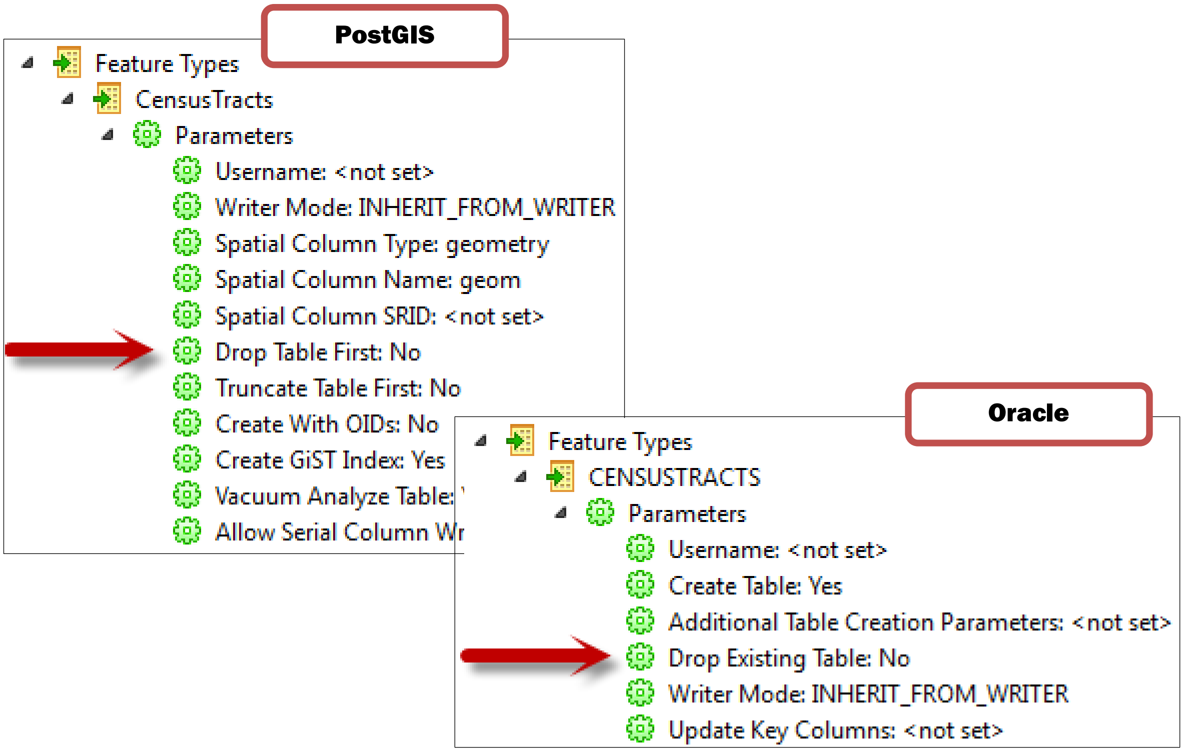
Writer Mode: INSERT

3) Set Feature Type Parameters (CensusTracts)
Locate the Feature Types entry for the database writer in the Workbench Navigator window.

Expand the list and you should see entries for CensusTracts, CountySubdivisions, and VotingDivisions. These are the tables that will be created when the workspace is run.



- CensusTracts/CountySubdivisions/VotingDivisions
- MyDatabase [POSTGIS]
  - Coordinate System: <not set>
  - Parameters
  - Feature Types
    - CensusTracts
    - CountySubdivisions
    - VotingDivisions

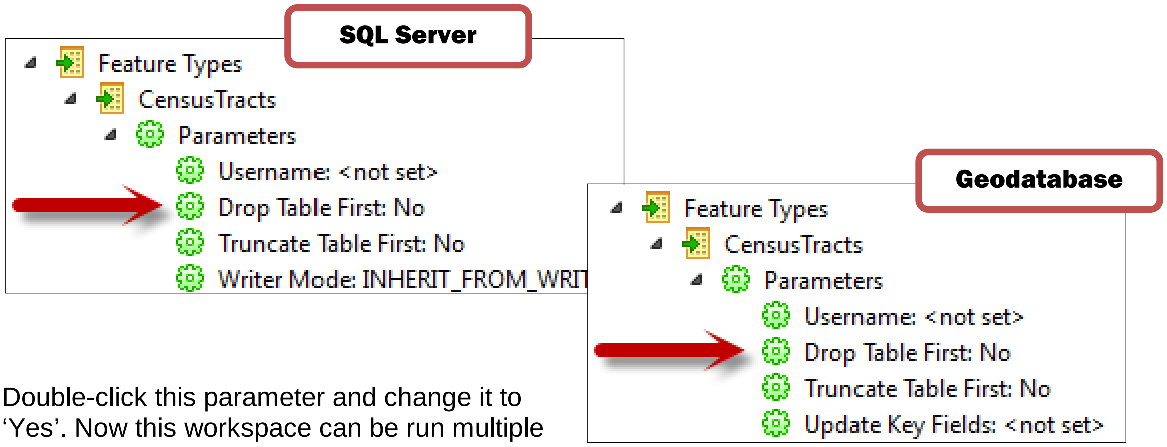


PostGIS

- Feature Types
  - CensusTracts
    - Parameters
      - Username: <not set>
      - Writer Mode: INHERIT_FROM_WRITER
      - Spatial Column Type: geometry
      - Spatial Column Name: geom
      - Spatial Column SRID: <not set>
      - Drop Table First: No
      - Truncate Table First: No
      - Create With OIDs: No
      - Create GiST Index: Yes
      - Vacuum Analyze Table: '
      - Allow Serial Column Wr

Oracle

- Feature Types
  - CENSUSTRACTS
    - Parameters
      - Username: <not set>
      - Create Table: Yes
      - Additional Table Creation Parameters: <not set>
      - Drop Existing Table: No
      - Writer Mode: INHERIT_FROM_WRITER
      - Update Key Columns: <not set>



SQL Server

- Feature Types
  - CensusTracts
    - Parameters
      - Username: <not set>
      - Drop Table First: No
      - Truncate Table First: No
      - Writer Mode: INHERIT_FROM_WRIT

Geodatabase

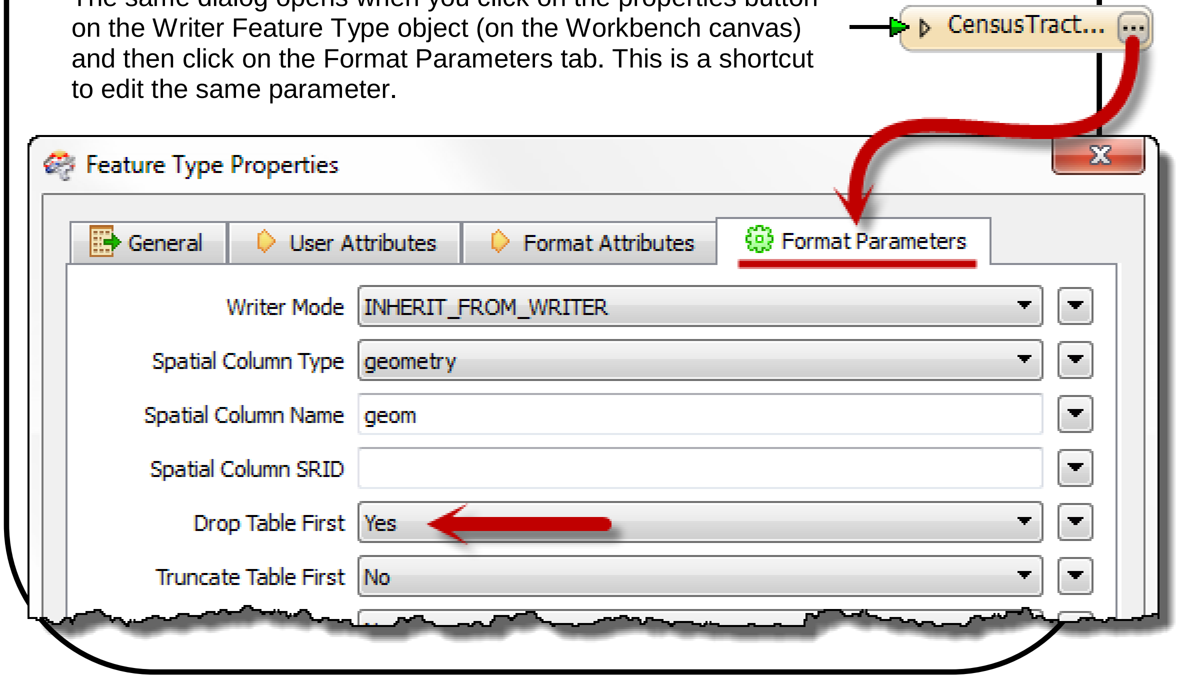
- Feature Types
  - CensusTracts
    - Parameters
      - Username: <not set>
      - Drop Table First: No
      - Truncate Table First: No
      - Update Key Fields: <not set>

Double-click this parameter and change it to ‘Yes’. Now this workspace can be run multiple times without ending up with multiple copies of the data.

Feature Type Parameters

Double-clicking a Feature Type parameter in the Navigator window opens up a dialog in which to edit that parameter.

The same dialog opens when you click on the properties button on the Writer Feature Type object (on the Workbench canvas) and then click on the Format Parameters tab. This is a shortcut to edit the same parameter.



Writer Mode: INHERIT_FROM_WRITER
Spatial Column Type: geometry
Spatial Column Name: geom
Spatial Column SRID:
Drop Table First: Yes
Truncate Table First: No

4) Set Feature Type Parameters (CountySubdivisions, VotingDivisions)

Repeat the previous step on the CountySubdivisions and VotingDivisions feature types.

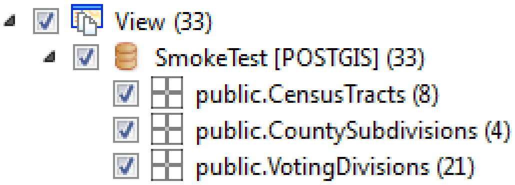
5) Save and Run Workspace

Save and then run the workspace.

The first time the workspace is run there may be warning messages that the tables to be dropped do not yet exist. This is just a warning and should not be a problem.

6) Inspect Output

Inspect the output in the FME Data Inspector. There should be three newly created tables to inspect.



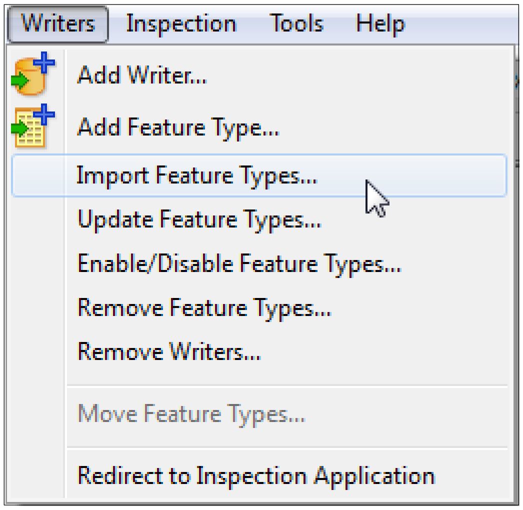
☑ View (33)
☑ SmokeTest [POSTGIS] (33)
☑ public.CensusTracts (8)
☑ public.CountySubdivisions (4)
☑ public.VotingDivisions (21)

Importing Table Schemas

Schemas can be imported when the tables to be written to already exist in the database.

By default, a new FME workspace creates database tables that are a duplicate of the incoming data. However, sometimes a user will wish to write to a database table that already exists.

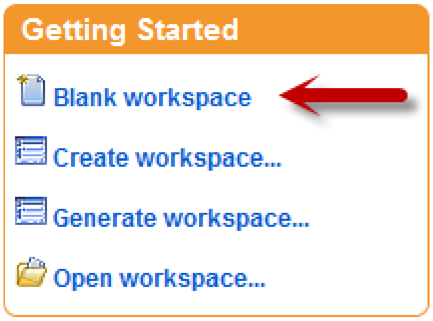
When a table already exists, the workspace should reflect the table schema (not the source), and this is achieved by importing the schema of that table from the database.



Import Feature Types
An Import Feature Types tool appears on the menubar of FME Workbench. It can be used once a writer has been added to a workspace, to import the schema from existing tables for use in that writer.

Follow these steps to create a workspace with imported feature type definitions:

1) Start FME Workbench.
Start FME Workbench. In the startup tab choose the option 'Blank workspace…'



2) Add Reader
In the new workspace select **Readers > Add Reader**. When prompted add the reader:

Reader Format:	ESRI Shape
Reader Dataset:	*C:\FMEData\Data\GovtBoundaries\CaldwellCounty\CensusTracts.shp* *C:\FMEData\Data\GovtBoundaries\CaldwellCounty\CountySubdivisions.shp* *C:\FMEData\Data\GovtBoundaries\CaldwellCounty\VotingDivisions.shp*

To select the files, click the File Browser button, browse to the correct folder, and then select all three Shape files in the folder. On accepting these, a reader will be added with three feature types.

3) Add Writer

Now select **Writers > Add Writer**. When prompted add the writer:

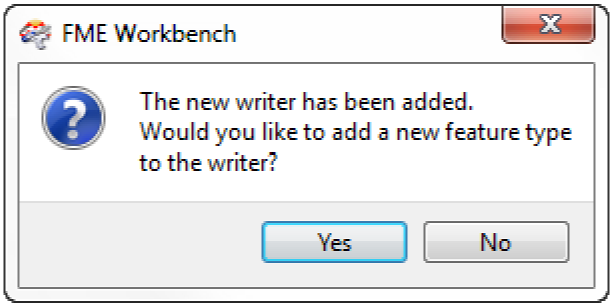
Writer Format: <database format of choice>

Again, choose your own database type, click the parameters button and then enter the connection parameters as tested earlier.

When prompted to add a new Feature Type, click **No**.

This is because the Feature Types will be imported, not added manually.

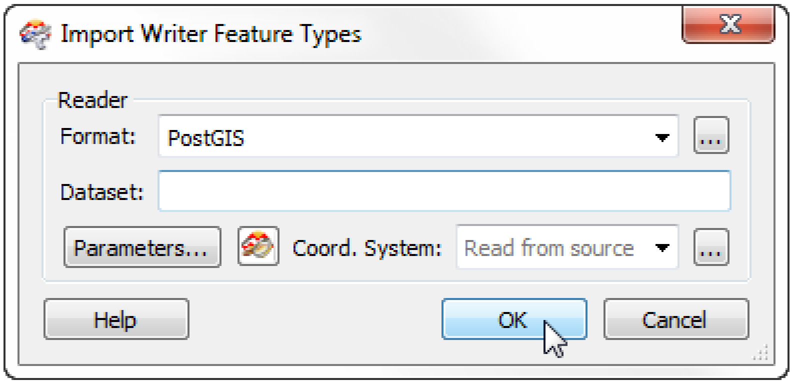
It might look like nothing has happened to the translation, but this is not the case. There are no Feature Types in the canvas, but a Writer is added in the Navigator.


FME Workbench

The new writer has been added. Would you like to add a new feature type to the writer?

4) Import Feature Types

Now select **Writers > Import Feature Types**.


Import Writer Feature Types

Reader

Format: PostGIS

Dataset:

Coord. System: Read from source

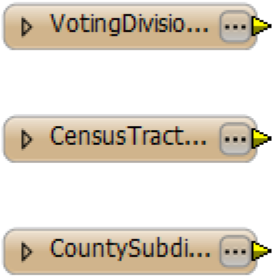
FME will first ask where the Feature Types are to be imported from.

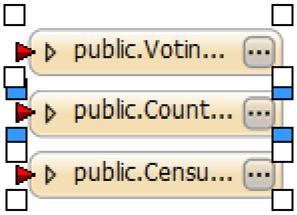
In this case they are the same location as the Writer dataset (though that might not always be true) so the dataset is correct.

Click on the Parameters button and use the Table List parameter to select the three tables CensusTracts, CountySubdivisions, and VotingDivisions.

Click **OK** to close each of the open dialogs.

The schema definitions of the three selected tables are now added to the workspace.

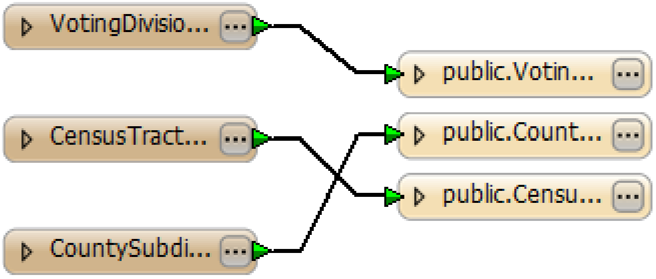




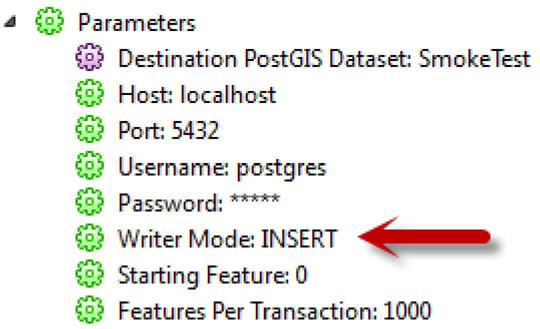
Geodatabase

When you import a table schema from ArcSDE, FME will include an Objectid field. Having this does not change the output results, but you may see warning messages in the log.

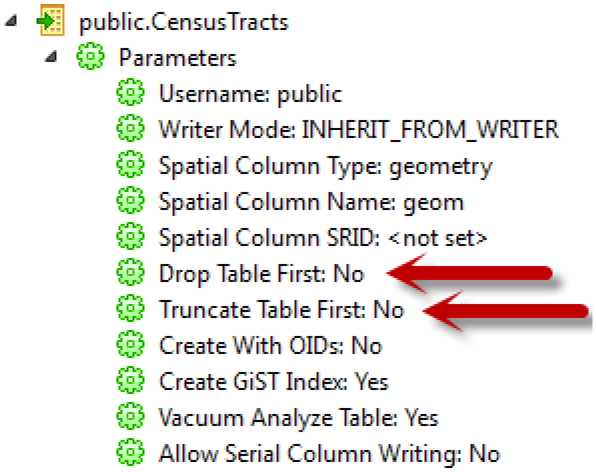
5) Connect Types
Connect the Reader and Writer Feature Types – be sure to get them connected to the correct ones as the order of the types may be different on the Reader and Writer!



6) Check Parameters
Check that the Writer Mode parameter is set to INSERT, and that each table has the "Drop" and "Truncate" options set to No (as this is a different county, we wish to add data, not replace it)



Parameters
- Destination PostGIS Dataset: SmokeTest
- Host: localhost
- Port: 5432
- Username: postgres
- Password: *****
- Writer Mode: INSERT
- Starting Feature: 0
- Features Per Transaction: 1000

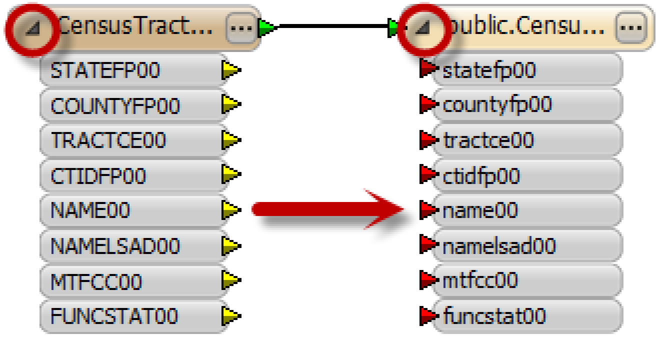


public.CensusTracts
- Parameters
  - Username: public
  - Writer Mode: INHERIT_FROM_WRITER
  - Spatial Column Type: geometry
  - Spatial Column Name: geom
  - Spatial Column SRID: <not set>
  - Drop Table First: No
  - Truncate Table First: No
  - Create With OIDs: No
  - Create GiST Index: Yes
  - Vacuum Analyze Table: Yes
  - Allow Serial Column Writing: No

7) Map Schema
Depending on the database being used, there may be a problem of non-mapping attributes.

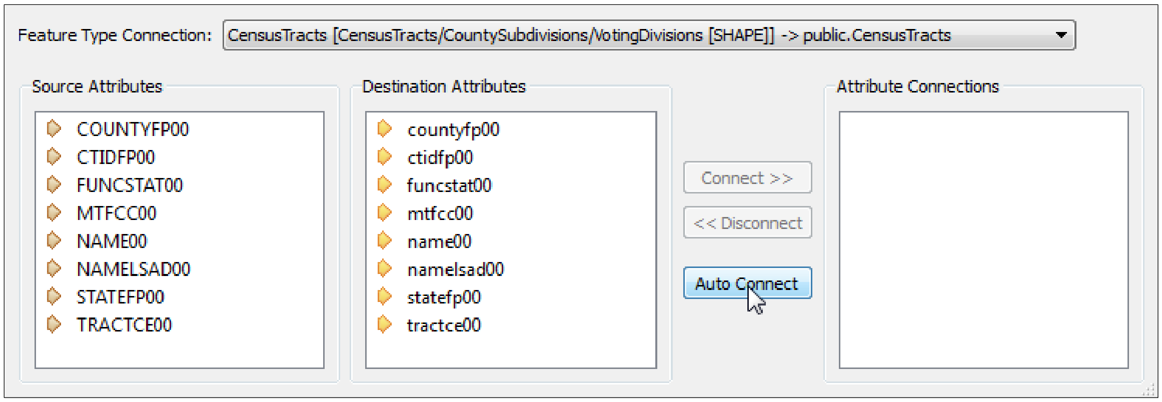
Expand the attribute list on reader and writer feature types by clicking the expand button.

If the attributes are upper case on the input, but lower case on the output, then they will not be transferred without being properly mapped.



If this problem occurs, select the menu item **View > Windows > Attribute Connections**.

Choose the Feature Type connection for CountySubdivisions. Click on **Auto Connect**. Repeat for each Feature connection.



Feature Type Connection: CensusTracts [CensusTracts/CountySubdivisions/VotingDivisions [SHAPE]] -> public.CensusTracts

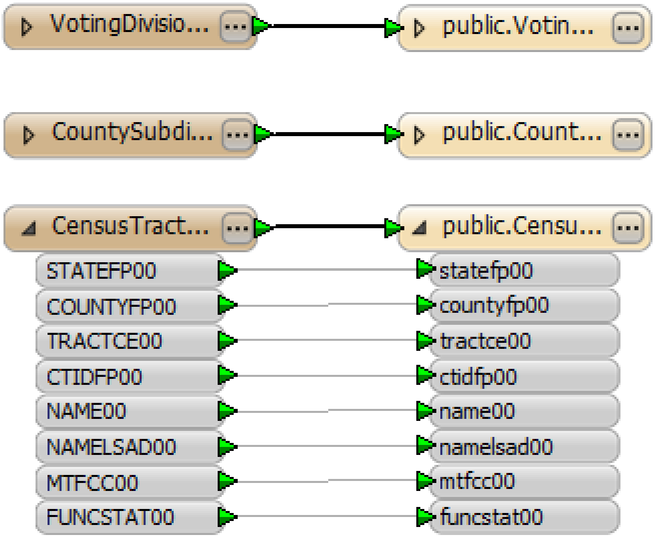
Source Attributes

- COUNTYFP00
- CTIDFP00
- FUNCSTAT00
- MTFCC00
- NAME00
- NAMELSAD00
- STATEFP00
- TRACTCE00

Destination Attributes

- countyfp00
- ctidfp00
- funcstat00
- mtfcc00
- name00
- namelsad00
- statefp00
- tractce00

Attribute Connections

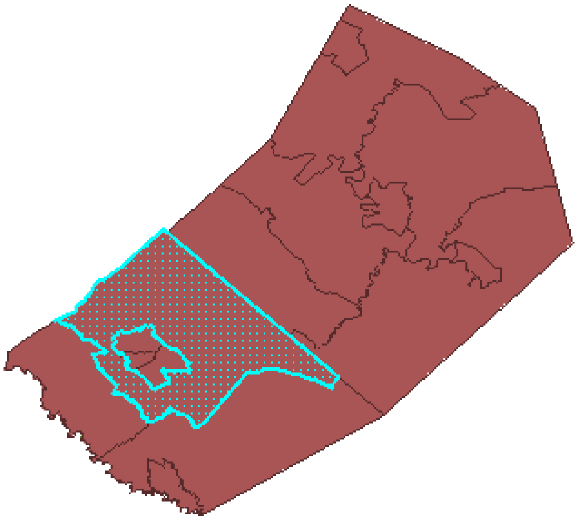


Because it's a simple matter of case-matching, FME is able to map the attributes easily.

8) Save and Run Workspace
Save the workspace and then run it.

Inspect the output to prove that there are now two counties stored in the database tables, and that they each have the correct attribute values.

***NB:** If the workspace is run more than once then new data will be added repeatedly!*



Reading from a Spatial Database

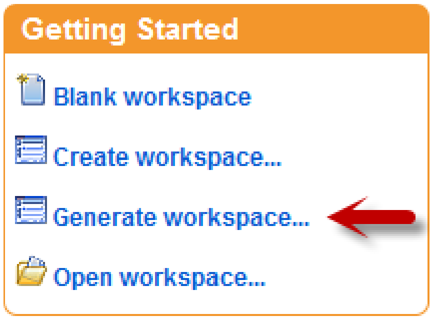
Once data is in the database, it can be read back.

Reading Data

Apart from the connection parameters, reading from a database is no more difficult that reading from a file-based dataset.

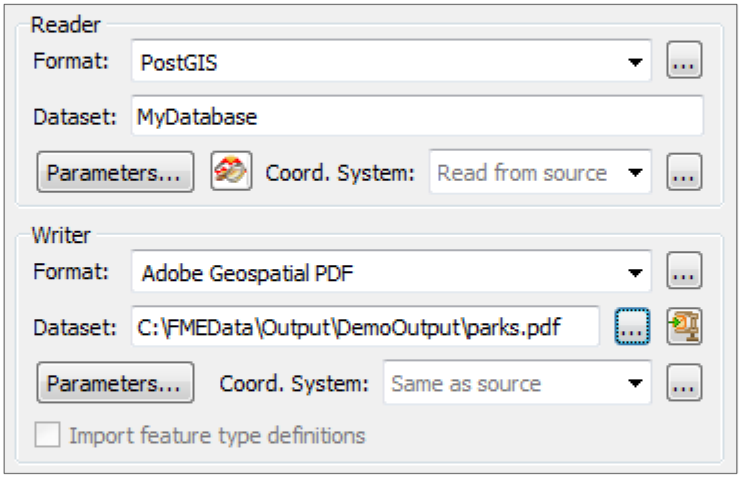
Follow these steps to read the parks data from your database:

1) Start FME Workbench.
Start FME Workbench. In the startup tab choose the option to 'Generate workspace…'



When the Generate Workspace dialog opens, fill in the fields as follows:

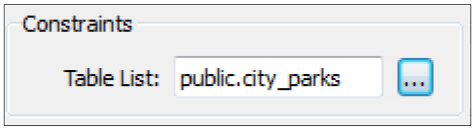
Reader Format:	<database format of choice>
Table:	*city_parks*
Writer Format:	Adobe Geospatial PDF
Writer Dataset:	*C:\FMEData\Output\DemoOutput\parks.pdf*



Reader
Format: PostGIS
Dataset: MyDatabase
Parameters... Coord. System: Read from source

Writer
Format: Adobe Geospatial PDF
Dataset: C:\FMEData\Output\DemoOutput\parks.pdf
Parameters... Coord. System: Same as source
□ Import feature type definitions

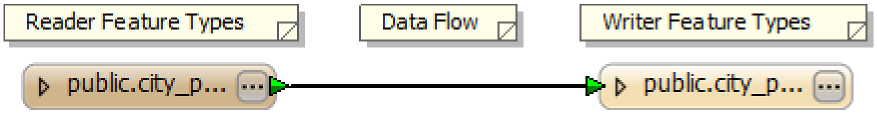
For the reader format, choose your own database type, click the parameters button and then enter the connection parameters as tested earlier.



Constraints
Table List: public.city_parks

At the same time, be sure to select the city_parks table from the available list.

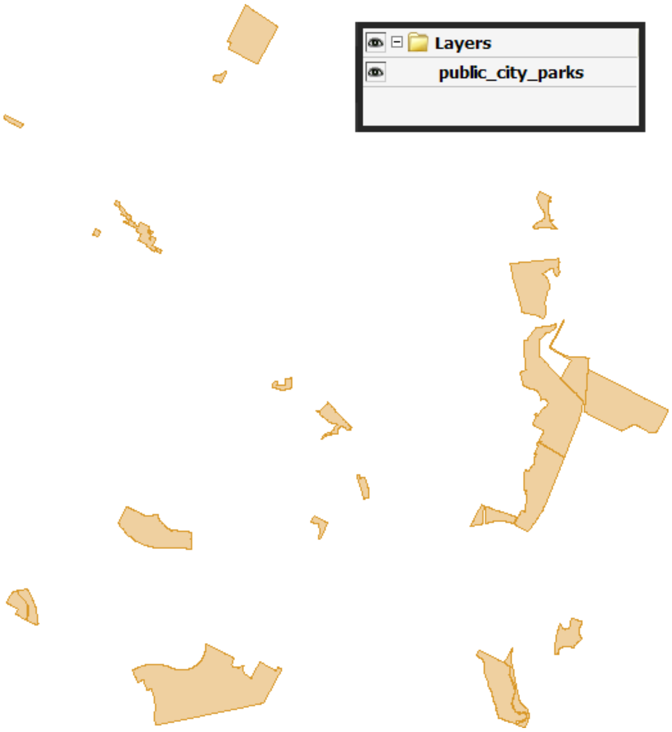
Click OK to create the workspace. It will look something like this:



2) Save and Run Workspace
Save and then run the workspace.

Data will be extracted from the database and written to a PDF file.

3) Inspect Output
Use Adobe Reader to view the contents of the PDF file and confirm that it contains the correct output.

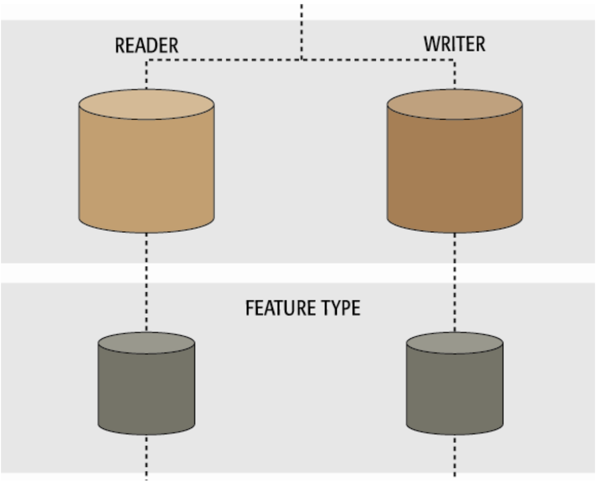


Controlling Database Reading

As with writers, various FME parameters exist to provide detailed control when reading a spatial database.

Just as there are different components to an FME translation, each component has its own set of parameters to control it.

So – when reading data - there are parameters to control the Reader (how FME handles the database itself) and also parameters to control the Feature Types (how FME handles individual tables).



Remember!

Reader/Writer	=	*Database*
Feature Type	=	*Table*

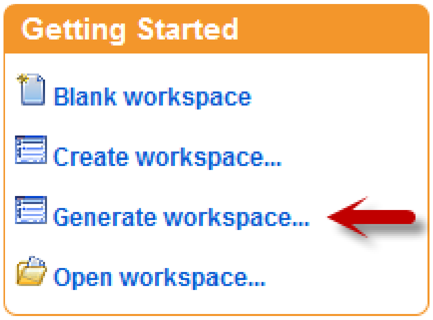
Reader Parameters
Readers are represented in the Navigator window of Workbench (by default the left-hand window) by a small, orange icon:

Clicking the little expand icon next to the Reader reveals all of the parameters associated with the reader. Each different format of Reader has its own set of parameters, many of them to control the connection parameters for the database.

Feature Type Parameters
Feature Types are represented in the Navigator window of Workbench by a slightly different orange icon:

Expanding the Feature Type reveals all the parameters associated with it. Again, each different format has its own set of feature type parameters.

Follow these steps to practice using Reader and Feature Type parameters:



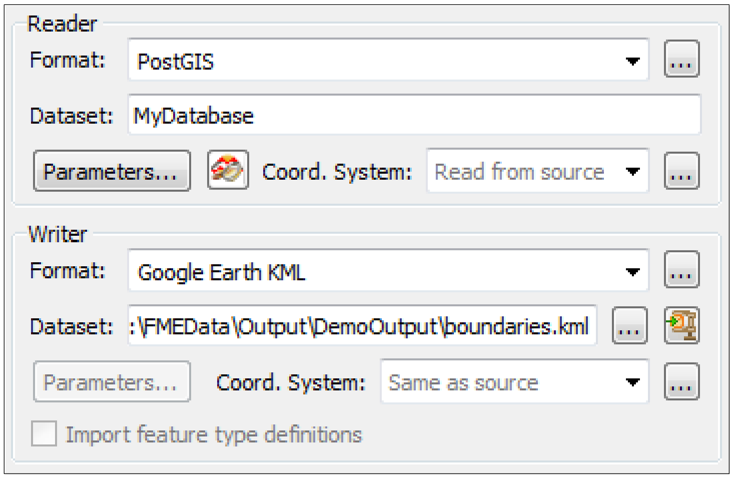
Getting Started

1) Start FME Workbench.
Start FME Workbench. In the startup tab choose the option to 'Generate workspace…'

When the Generate Workspace dialog opens, fill in the fields as follows:

Reader Format:	<database format of choice>
Tables:	*VotingDivisions, CensusTracts, CountySubdivisions*
Writer Format:	Google Earth KML
Writer Dataset:	*C:\FMEData\Output\DemoOutput\boundaries.kml*

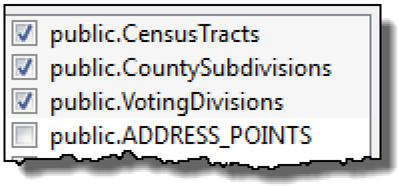
For the reader format, choose your own database type, click the Parameters button and then enter the connection parameters as tested earlier.



Reader
Format: PostGIS
Dataset: MyDatabase
Coord. System: Read from source

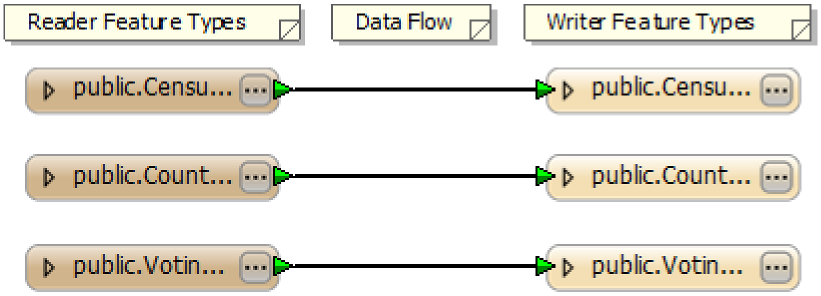
Writer
Format: Google Earth KML
Dataset: :\FMEData\Output\DemoOutput\boundaries.kml
Coord. System: Same as source
□ Import feature type definitions

At the same time, be sure to select the required tables from the available list.



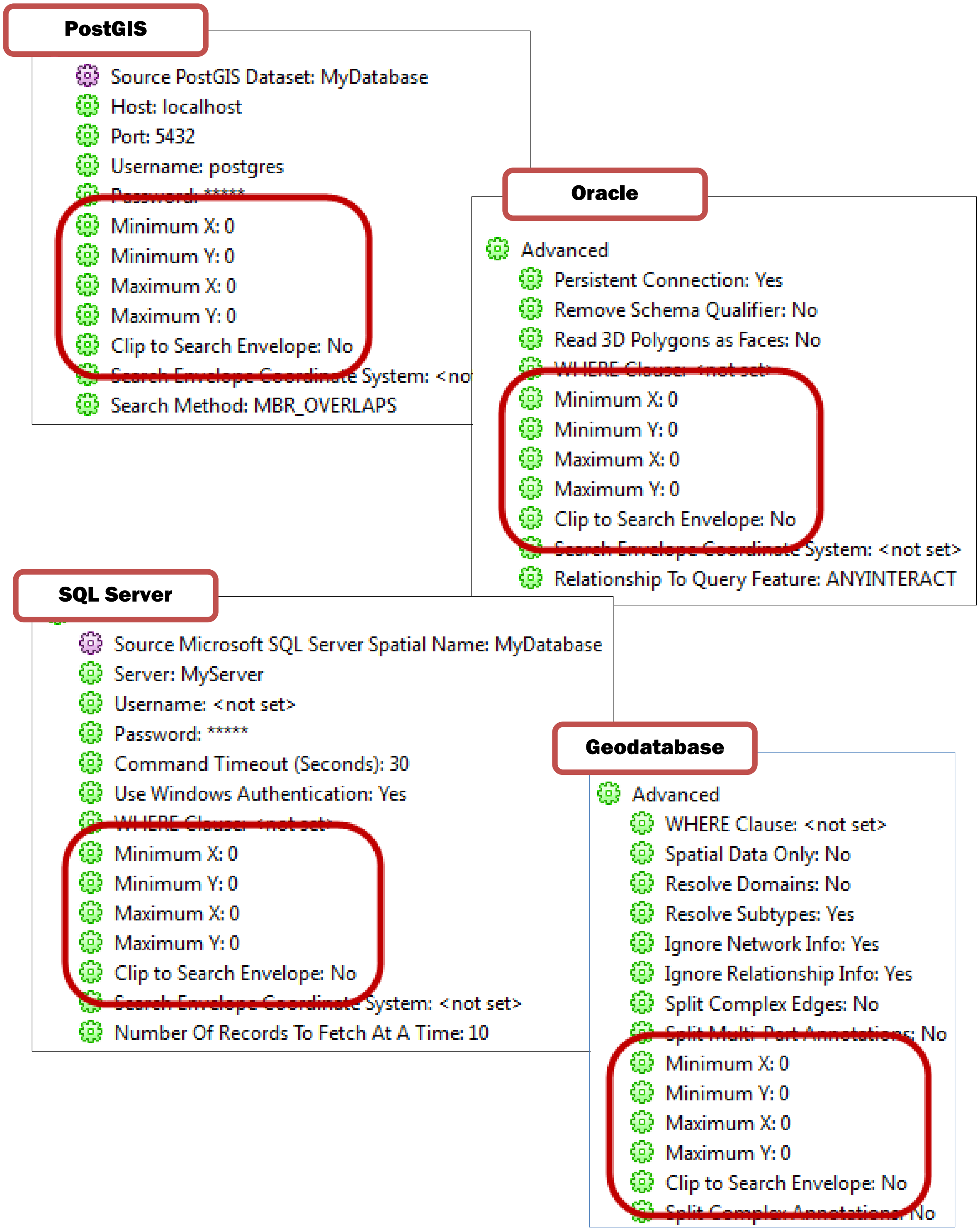
☑ public.CensusTracts
☑ public.CountySubdivisions
☑ public.VotingDivisions
□ public.ADDRESS_POINTS

Click OK to create the workspace. It will look something like this:



2) Set Reader Parameters
Locate the database reader in the Workbench Navigator window.

Expand the list of parameters, and locate the minimum and maximum coordinate parameters, and the 'Clip to Search Envelope' parameter. Their positions will vary according to the database format being used:



PostGIS

- Source PostGIS Dataset: MyDatabase
- Host: localhost
- Port: 5432
- Username: postgres
- Password: *****
- Minimum X: 0
- Minimum Y: 0
- Maximum X: 0
- Maximum Y: 0
- Clip to Search Envelope: No
- Search Envelope Coordinate System: <no
- Search Method: MBR_OVERLAPS

Oracle

- Advanced
  - Persistent Connection: Yes
  - Remove Schema Qualifier: No
  - Read 3D Polygons as Faces: No
  - WHERE Clause: <not set>
  - Minimum X: 0
  - Minimum Y: 0
  - Maximum X: 0
  - Maximum Y: 0
  - Clip to Search Envelope: No
  - Search Envelope Coordinate System: <not set>
  - Relationship To Query Feature: ANYINTERACT

SQL Server

- Source Microsoft SQL Server Spatial Name: MyDatabase
- Server: MyServer
- Username: <not set>
- Password: *****
- Command Timeout (Seconds): 30
- Use Windows Authentication: Yes
- WHERE Clause: <not set>
- Minimum X: 0
- Minimum Y: 0
- Maximum X: 0
- Maximum Y: 0
- Clip to Search Envelope: No
- Search Envelope Coordinate System: <not set>
- Number Of Records To Fetch At A Time: 10

Geodatabase

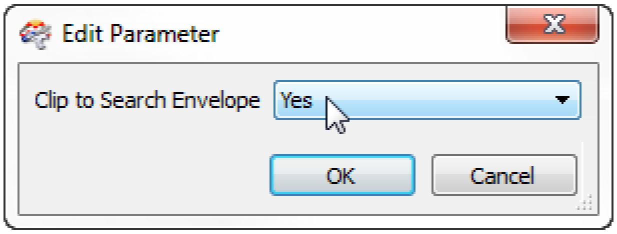
- Advanced
  - WHERE Clause: <not set>
  - Spatial Data Only: No
  - Resolve Domains: No
  - Resolve Subtypes: Yes
  - Ignore Network Info: Yes
  - Ignore Relationship Info: Yes
  - Split Complex Edges: No
  - Split Multi-Part Annotations: No
  - Minimum X: 0
  - Minimum Y: 0
  - Maximum X: 0
  - Maximum Y: 0
  - Clip to Search Envelope: No
  - Split Complex Annotations: No

Double-click each of the min/max X/Y parameters in turn to set a search envelope for the reader. Set the following values:

Minimum X	-97.4
Minimum Y	30.0
Maximum X	-97.2
Maximum Y	30.2

Also set the Clip to Search Envelope parameter to ‘Yes’.

Now when the workspace is run, only a certain area of data will be read, and clipped to the defined envelope.



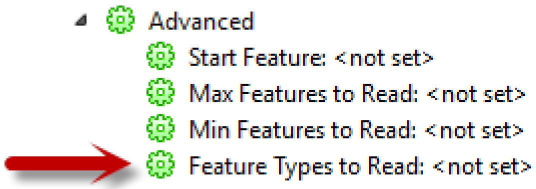
Clip to Search Envelope: Yes

3) Set Reader Parameters

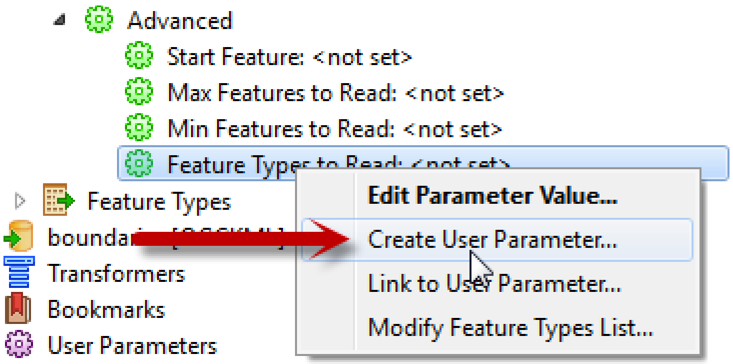
Another reader parameter that is most useful for databases is “Feature Types to Read”.

This parameter lets you choose which tables to read from a database, whenever the workspace is run.

Expand the advanced Reader parameters for the database reader and locate the parameter labeled “Feature Types to Read”



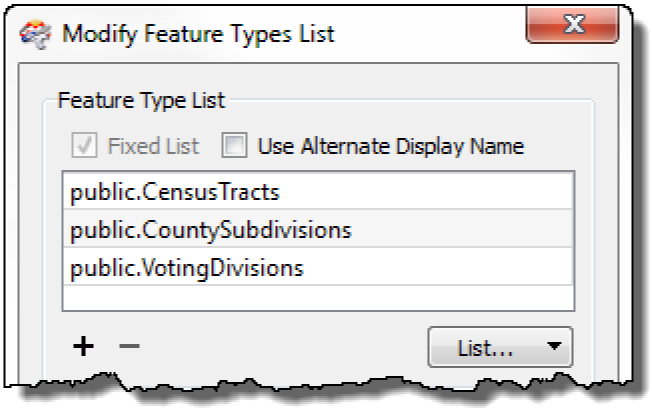
Start Feature: <not set>
Max Features to Read: <not set>
Min Features to Read: <not set>
Feature Types to Read: <not set>


Advanced
Start Feature: <not set>
Max Features to Read: <not set>
Min Features to Read: <not set>
Feature Types to Read: <not set>
Feature Types
bounda... [OGCKML]
Transformers
Bookmarks
User Parameters
Edit Parameter Value...
Create User Parameter...
Link to User Parameter...
Modify Feature Types List...

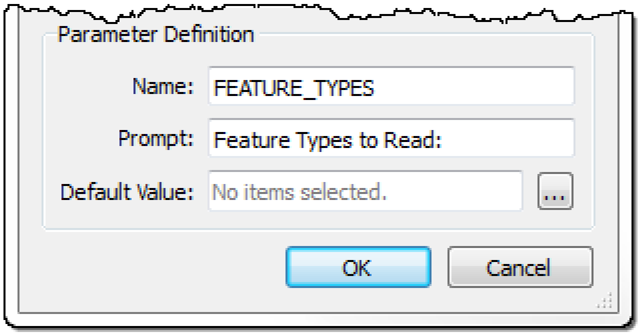
Right-click the parameter and choose the option **Create User Parameter**

Creating a “User Parameter” means that the workspace user is prompted at run-time to select the tables to be processed.

The dialog that opens up will show a list of tables that are currently in the workspace:



public.CensusTracts
public.CountySubdivisions
public.VotingDivisions


Parameter Definition

Name: FEATURE_TYPES

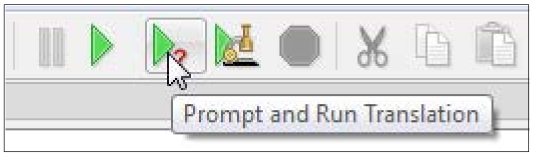
Prompt: Feature Types to Read:

Default Value: No items selected.

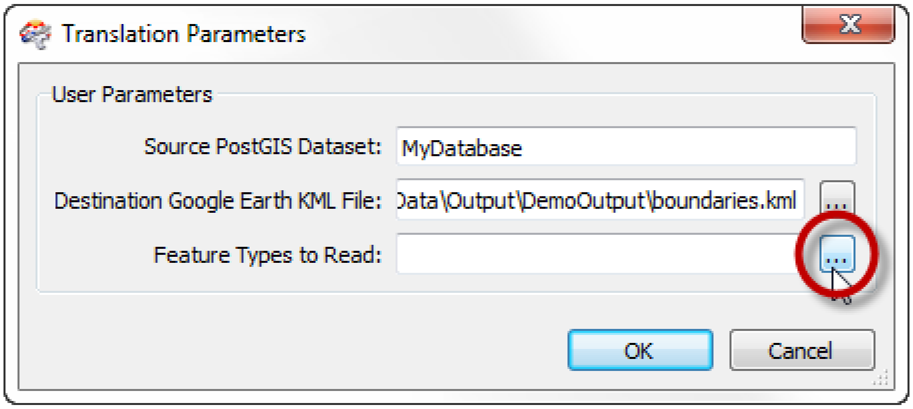
…and will also show a definition of what the user will be prompted with.

Simply click **OK** to accept the default settings.

Now run the workspace using the Prompt and Run Translation option on the toolbar (or menubar)



In the Translation Parameters dialog, click the browse button next to the Feature Types to Read parameter, and choose a selection of tables to be read.


Translation Parameters

User Parameters

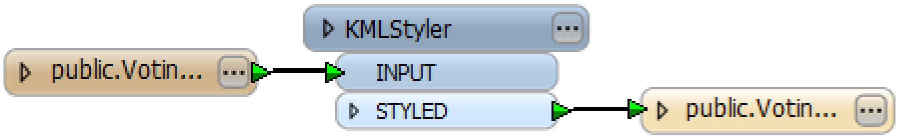
Source PostGIS Dataset: MyDatabase

Destination Google Earth KML File: Data\Output\DemoOutput\boundaries.kml

Feature Types to Read:

The translation is now carried out, reading your chosen selection of tables, and clipping them to the previously defined envelope. If you have Google Earth installed, open the output dataset to confirm the result of the translation.

A KMLStyler transformer can be used to give color and style to features that are to be viewed in Google Earth.



Basic Database Updates

Most users who write to a database will want to update data at some point.

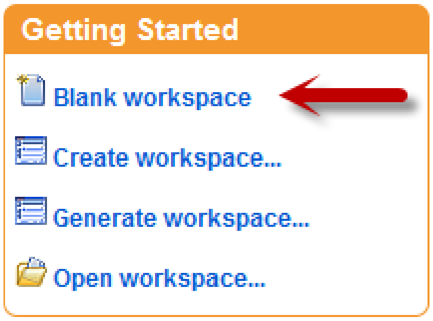
Updates to a database can be as simple as dropping or truncating the table (using feature type parameters) before loading new data.

However, in most cases, updates will be more specific, and not wish to touch unrelated data.

FME Update Modes
FME can update all data in a database, or all data in a specific table; it can even carry out updates on single features, while inserting or deleting others.

Follow these steps to update individual features within a dataset:

1) Start FME Workbench.
Start FME Workbench. In the startup tab choose the option 'Blank workspace…'



2) Add Reader
In the new workspace select **Readers > Add Reader**. When prompted add the Reader:

Reader Format:	ESRI Shape
Reader Dataset:	*C:\FMEData\Data\GovtBoundaries\CaldwellCounty\CensusTracts.shp*

The scenario here is that the CensusTracts dataset for Caldwell County has been edited and needs to be updated in the spatial database.

3) Add Writer

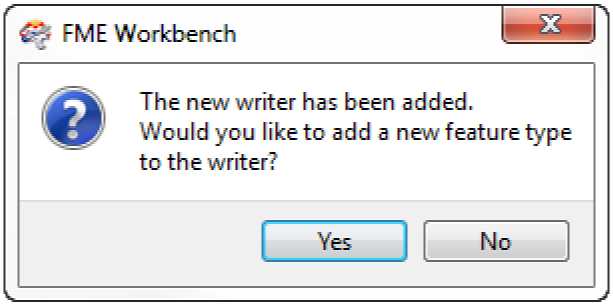
Now select **Writers > Add Writer**. When prompted add the writer:

Writer Format: <database format of choice>

Again, choose your own database type, click the parameters button and then enter the connection parameters as tested earlier.

When prompted to add a new Feature Type, click **No**.

This is because the Feature Types will be imported, not added manually.

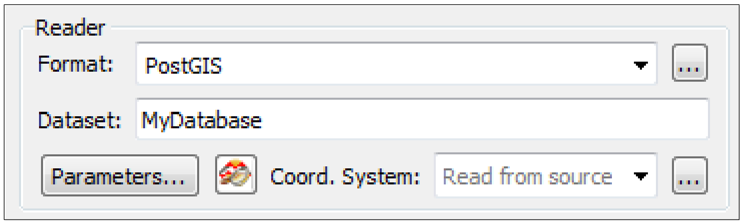


FME Workbench

The new writer has been added.
Would you like to add a new feature type to the writer?

4) Import Feature Types

Now select **Writers > Import Feature Types**.



Reader

Format: PostGIS

Dataset: MyDatabase

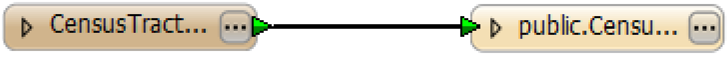
Coord. System: Read from source

FME will first ask where the Feature Types are to be imported from.

In this case they are the same location as the Writer dataset (though that might not always be true) so the dataset is correct.

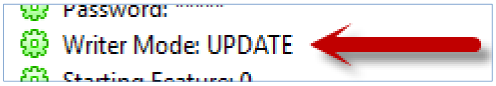
Click the Parameters button and use the Table List parameter to select CensusTracts.

Connect the newly imported Writer Feature Type to the existing reader feature type.



5) Set Writer Mode

This time set the Writer Mode parameter to UPDATE.



Password: ********

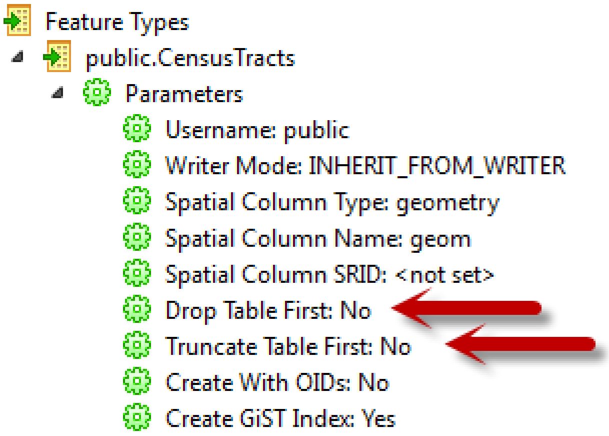
Writer Mode: UPDATE

Starting Feature: 0

6) Set Feature Type Parameters

Now check the Feature Type parameters and ensure the “Drop” and “Truncate” options are set to No.

This is important because we wish to replace the Caldwell County data, but without deleting the table or the Bastrop County data that already exists within it.

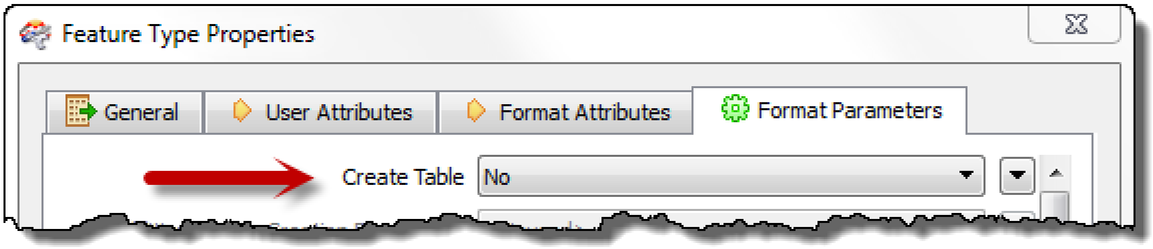


Feature Types
- public.CensusTracts
  - Parameters
    - Username: public
    - Writer Mode: INHERIT_FROM_WRITER
    - Spatial Column Type: geometry
    - Spatial Column Name: geom
    - Spatial Column SRID: <not set>
    - Drop Table First: No
    - Truncate Table First: No
    - Create With OIDs: No
    - Create GiST Index: Yes

Oracle

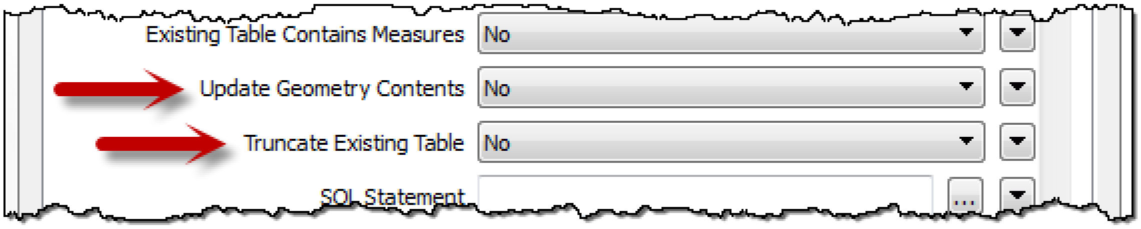
For Oracle, the functionality is controlled slightly differently here.

Firstly set **Create Table** to No.
Obviously the table already exists and we do not want to try and create it.



Create Table: No

Then set **Truncate Existing Table** to No, and also set **Update Geometry Contents** to No.
Geometry contents should not be updated because only attributes are being changed.

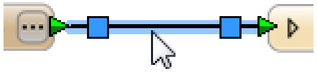


Existing Table Contains Measures: No
Update Geometry Contents: No
Truncate Existing Table: No
SQL Statement:

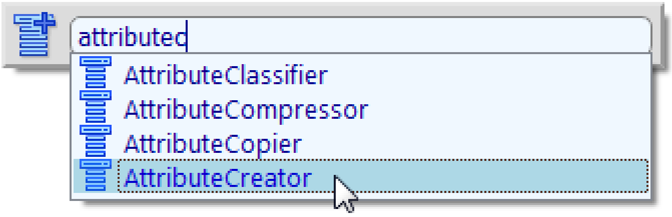
7) Add Transformer

To actually make a change to the source data (to prove the update worked) one of the attribute values can be changed with an *AttributeCreator* transformer.

In the Workbench canvas click on the connection between the Reader and Writer Feature Types.

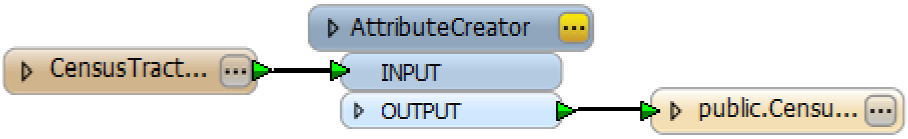


Start typing the word “attributec”. You will see the Quick Add list of matching transformers appear beneath.



attributec

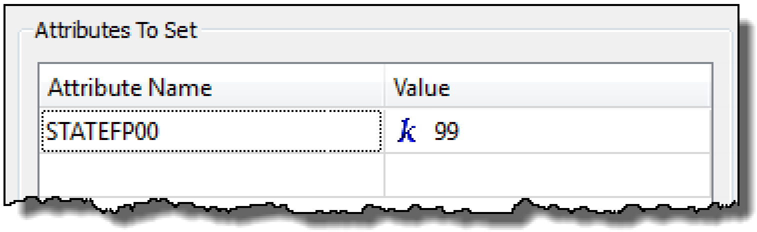
Select the transformer named *AttributeCreator.* The transformer is dropped automatically into place between the Reader and Writer.



8) Set Parameters

The yellow-colored icon on the top-right of the *AttributeCreator* transformer indicates there are parameters for this transformer that require checking or updating.

Click on the icon. This will open a parameters dialog.



Attributes To Set

Attribute Name	Value
STATEFP00	99

Select STATEFP00 as the attribute to be set.

Enter a value of 99 as the new attribute value.

***NB:** Don't enter a value greater than 99, as the output field is only defined as a two digit number.*

Leave this dialog open for the next step.

9) Set Database Action

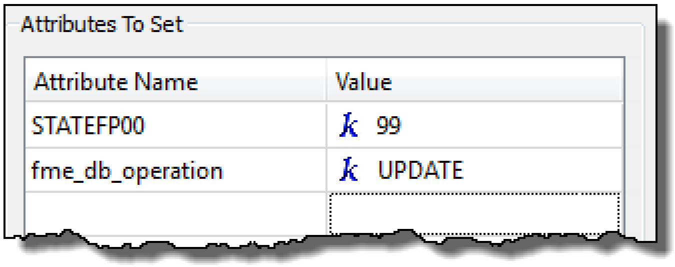
To specify what database action to carry out on each feature (insert, update, or delete) FME provides what is known as a 'Format Attribute' or 'FME Attribute'.

These are attributes that relate to data structure, not user-defined feature attributes. The *AttributeCreator* transformer is one way to create these attributes.

In the parameters dialog for *AttributeCreator*, add this format attribute:

Attribute Name	**Value**
fme_db_operation	UPDATE

The name fme_db_operation **must** be in lower case.



Attributes To Set

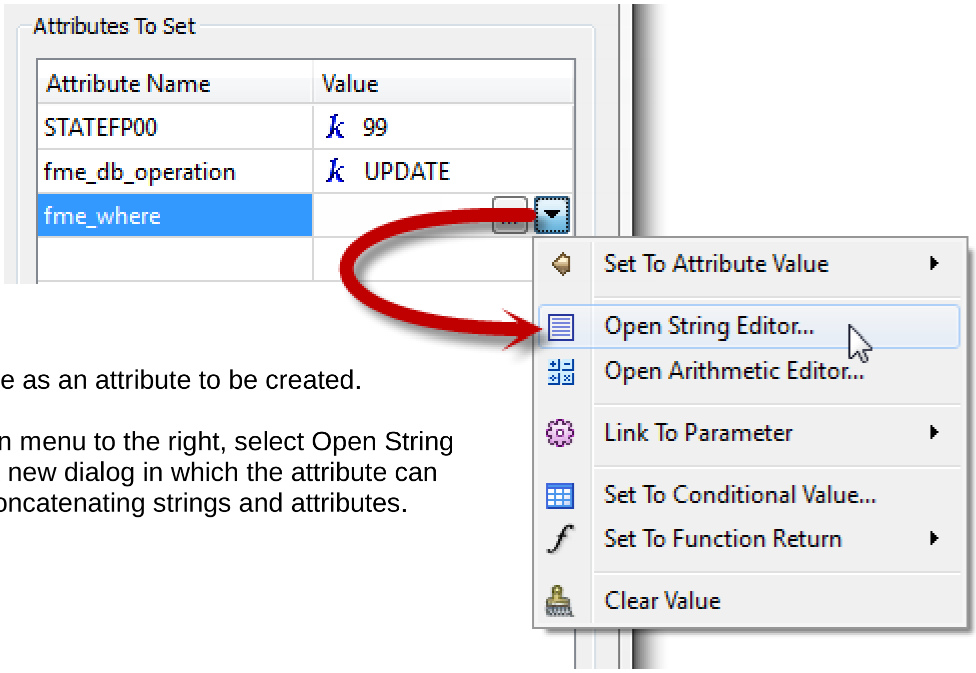
Attribute Name	Value
STATEFP00	99
fme_db_operation	UPDATE

Again, leave this dialog open for the next step.

10) Set WHERE Clause

When the database operation is an UPDATE or DELETE it is necessary to define a WHERE clause to specify which features should be updated or deleted.

Again this is a format attribute that can be created with the AttributeCreator transformer.



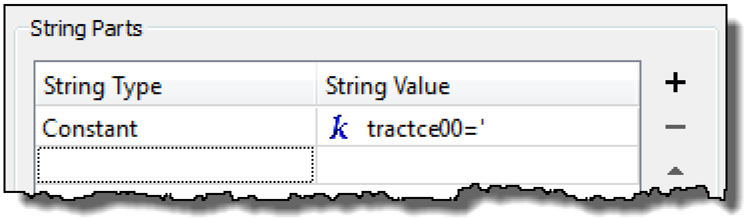
Attributes To Set

Attribute Name	Value
STATEFP00	99
fme_db_operation	UPDATE
fme_where	

Enter fme_where as an attribute to be created.

In the drop-down menu to the right, select Open String Editor to open a new dialog in which the attribute can be defined by concatenating strings and attributes.

In the Text Editor dialog, click in the first row, under the String Type heading.



String Parts

String Type	String Value
Constant	tractce00='

Select ‘Constant’ as the string type.

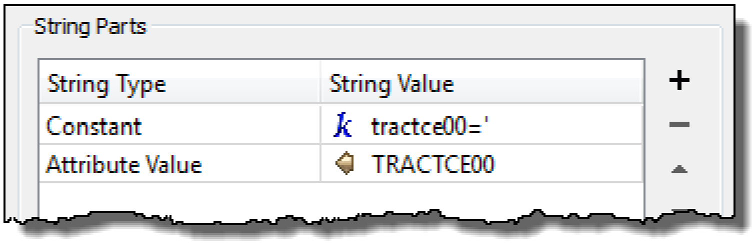
For String Value, enter: `tractce00='`

This is the first part of the fme_where clause.

tractce00 is the attribute name by which to match features. If it is upper case in your database then enter it as TRACTCE00 instead. Don’t miss the single quote ‘ character at the end of the string, as this is the opening quote around the attribute value.

In the second row, select 'Attribute Value' as the string type.

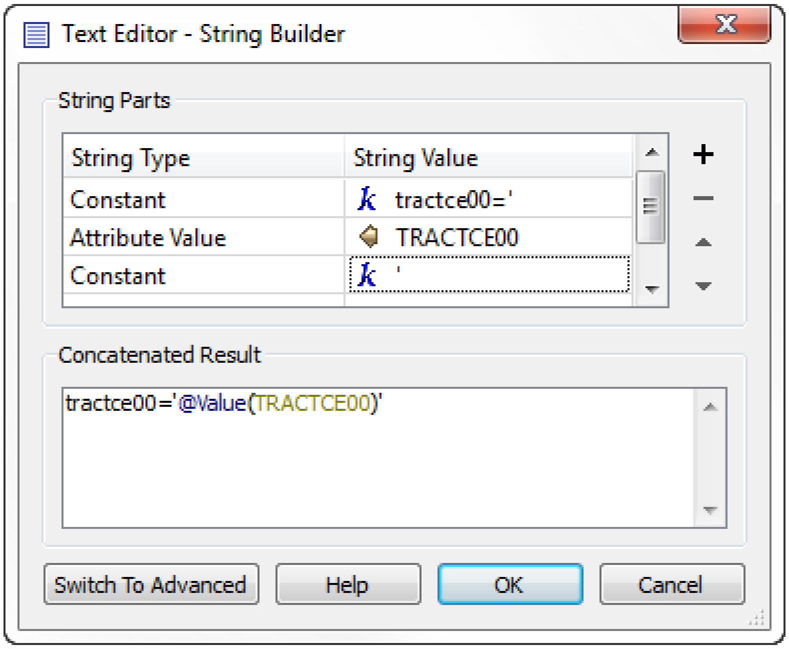
Now select TRACTCE00 from the list of attributes.



String Parts

String Type	String Value
Constant	tractce00='
Attribute Value	TRACTCE00

Finally, in the third row, select Constant for the string type, and enter a closing single quote ' character.



Text Editor - String Builder

String Parts

String Type	String Value
Constant	tractce00='
Attribute Value	TRACTCE00
Constant	'

Concatenated Result

tractce00='@Value(TRACTCE00)'

The Concatenated Result field now looks like this: `tractce00='@Value(TRACTCE00)'`

Click **OK** to close the Text Editor, then again to close the AttributeCreator parameters dialog.

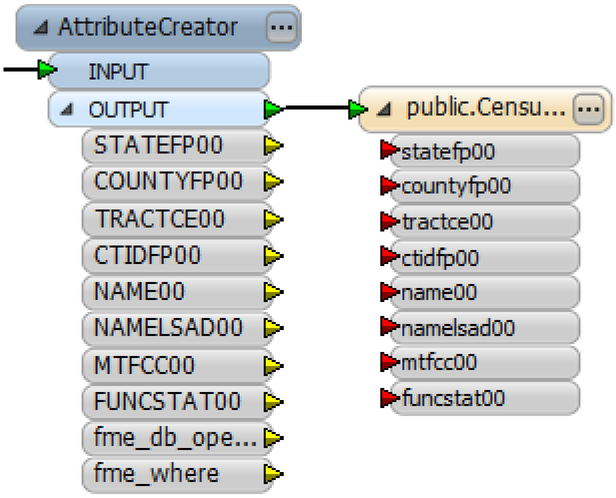
> *If an attribute (here TRACTCE00) is stored as a numeric value then you don't need the quotes around it. Because (in my PostGIS database) it is a Varchar, then the quotes are necessary.*

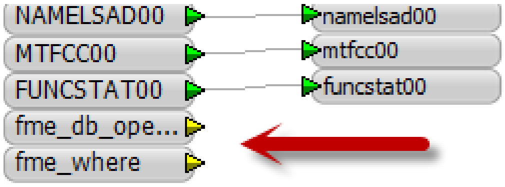
12) Map Schema

If the attribute mapping is missing again, then open **View > Windows > Attribute Connections**.

Select the Feature connection for AttributeCreator:OUTPUT -> public.CensusTracts

Click on **Auto Connect** to reconnect the attributes.



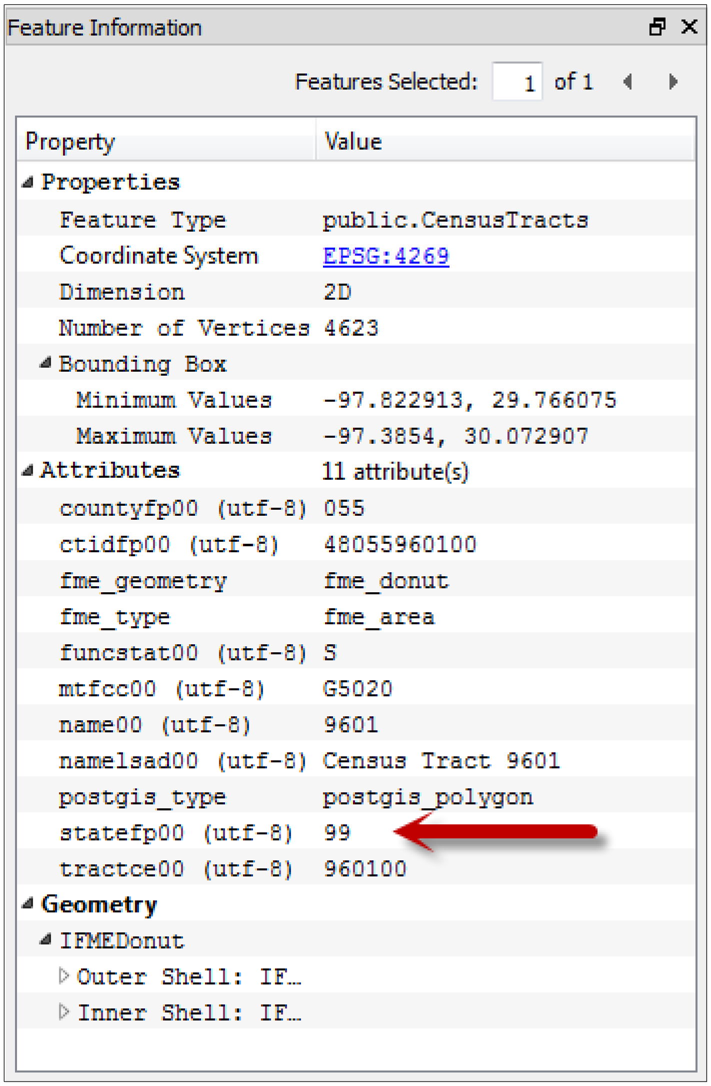


***NB:** It doesn't matter that fme_db_operation and fme_where do not get connected to anything. They just need to exist!*

13) Save and Run Workspace
Save the workspace and then run it.

Inspect the CensusTract table to prove that:

a) The Bastrop County data still exists
b) The Caldwell County data exists, and that there is still only one copy of it
c) The statefp00 attribute has been updated to 99



Feature Information

Features Selected: 1 of 1

Property	Value
◢ Properties	
Feature Type	public.CensusTracts
Coordinate System	EPSG:4269
Dimension	2D
Number of Vertices	4623
◢ Bounding Box	
Minimum Values	-97.822913, 29.766075
Maximum Values	-97.3854, 30.072907
◢ Attributes	11 attribute(s)
countyfp00 (utf-8)	055
ctidfp00 (utf-8)	48055960100
fme_geometry	fme_donut
fme_type	fme_area
funcstat00 (utf-8)	S
mtfcc00 (utf-8)	G5020
name00 (utf-8)	9601
namelsad00 (utf-8)	Census Tract 9601
postgis_type	postgis_polygon
statefp00 (utf-8)	99
tractce00 (utf-8)	960100
◢ Geometry	
◢ IFMEDonut	
▷ Outer Shell: IF…	
▷ Inner Shell: IF…	

***NB:** Because this workspace is carrying out an UPDATE, it can be run many times without creating multiple data copies.*

Troubleshooting

Here are some useful hints on how to troubleshoot problems with spatial databases in FME

Because many error messages come directly from the database application, it’s difficult to provide a common set of errors that might occur.

Generally the problems that do occur are caused by connection issues or bad data.

Connection Problems

If FME has problems the first test must be to try and connect with a native database application, such as (in the case of Oracle) Oracle’s SQL Developer or Toad™ (Tool for Oracle Application Developers). SQLPlus is not a reliable test!

Be sure to do this using the same user credentials! The log window will usually report which database it is trying to connect to, and what the user credentials are.

Try to find an error message in the log window that is a standard for that particular database (for example an ORA-xxxx error number) and use the database documentation to look up that particular error.

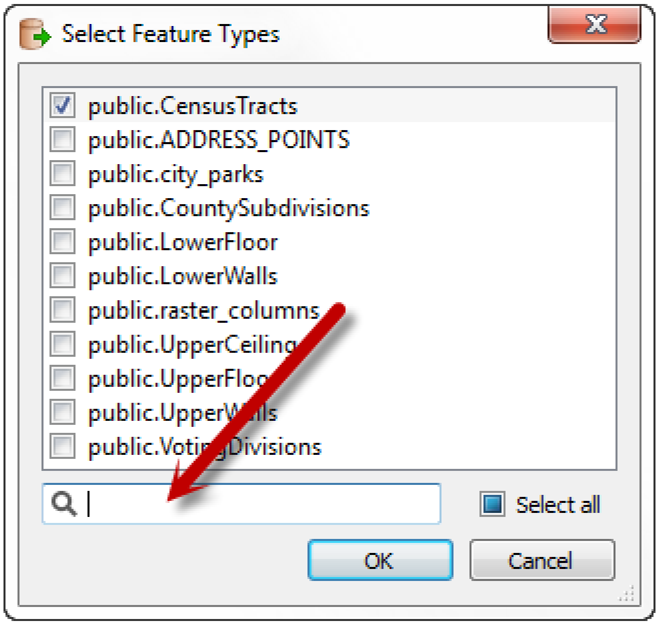
Be sure that you are using compatible components; for example 32-bit FME can connect to a 64-bit Oracle database, but needs the 32-bit client.

Reading Problems

A common reading issue is the list of tables returned by the parameters dialog.

If no tables show up (or FME reports an error) firstly ensure that there are actually tables with data in them. If you are using a spatial Reader in FME, then the data must have a spatial component for the table to appear. This may include the requirement to have metadata information defining spatial extents and coordinate systems.

Also check that there is no filter set in the dialog, which might prevent tables from appearing in the list:



☑ public.CensusTracts
□ public.ADDRESS_POINTS
□ public.city_parks
□ public.CountySubdivisions
□ public.LowerFloor
□ public.LowerWalls
□ public.raster_columns
□ public.UpperCeiling
□ public.UpperFloor
□ public.UpperWalls
□ public.VotingDivisions

▣ Select all

Writing Problems

Firstly, make sure there is enough room in the database to write the required data! The message that reports this might be quite obscure or misleading (e.g. “unable to create initial extent for tablespace”) so be aware of this potential problem.

When writing spatial data, sometimes FME will adjust it automatically, to ensure it conforms to the database specification. For example, duplicate vertices would be removed before data is written to an Oracle database. If the output is different in size to the input, this might be why.

Some databases have specific boundaries to the spatial data area. A common problem is to write data that falls outside these boundaries (again the warning message can be quite obscure). Check your Writer to see what the values are for any X/Y origin settings.

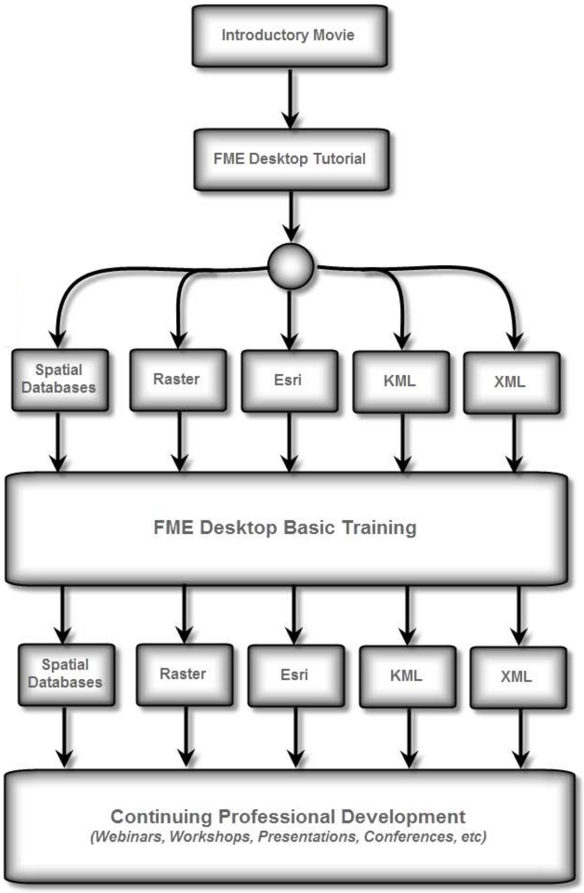
Common table creation problems occur when a table already exists, the table name is invalid, or the user has insufficient permissions. Sometimes a failure can occur when data is not compatible with the attribute type that it is being written to.

Some data types (such as Geodatabase relationship classes) can’t be created by FME, but can only be written to if they already exist. In such cases create the classes with a native application, and use the Import Feature Types tool to bring them into Workbench.

Finally, be aware of any dependencies and try to make sure that parent tables are written first. The easiest way to achieve this is to use a writer for each different table, and order the writers in the Navigator window.

What's Next?

This document is a basic introduction to using spatial databases with FME


Introductory Movie
FME Desktop Tutorial
Spatial Databases
Raster
Esri
KML
XML
FME Desktop Basic Training
Spatial Databases
Raster
Esri
KML
XML
Continuing Professional Development
(Webinars, Workshops, Presentations, Conferences, etc)

Next Step
The next step in the FME Database Pathway is to take an FME Desktop Basic Training course, in preparation for the full FME Database Training Course.

Further information on all training options is available on the Safe Software web site at www.safe.com/training

Many other resources for FME Desktop technical information can be located through the FMEpedia knowledgebase at http://fmepedia.safe.com