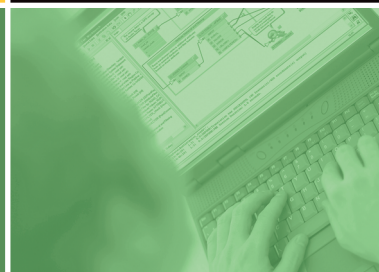


2009 FME International User Conference: Training Module  
Dynamic Workspaces and the Three R's



---

## Dynamic Workspaces

---

Dynamic Workspaces.....	2
Why Make Dynamic Workspaces? .....	2
What can be made Dynamic.....	2
Introduction to Dynamic Workspaces.....	3
Dynamic Workspaces .....	3
Data Distribution with Dynamic Workspaces .....	5
Traditional Workspaces .....	5
Dynamic Data Distribution.....	7
Create a Dynamic Workspace .....	7
Choosing the source feature types .....	8
Changing the Dynamic Destination Schema.....	9
Schema Readers.....	10
Configure the Schema Reader.....	11
Advanced Dynamic Schema Topics .....	12
Generic Writer .....	13
Generic Writer Format .....	13
Adding the Generic Writer.....	14
Advanced Generic Writer Configuration.....	16
Review.....	18
What You Should Have Learned.....	18

## Dynamic Workspaces

Dynamic workspaces are a way of providing maximum translation flexibility and minimizing the longer term maintenance of workspaces.

### Why Make Dynamic Workspaces?

Traditional FME Workspaces are very tightly bound to the source and destination schemas. In many cases this is what you want. But in certain situations the workspace needs to be more independent of the schemas.

A dynamic workspace breaks the dependence on the source and destination schema. Common applications for dynamic workspaces are:

- when a relatively simple data transformation is to be applied to all the data irrespective of schema (i.e. clipping, coordinate transformation)
- when the source schema is not predictable or well defined
- ad hoc data transformations
- when the destination format might vary
- long term maintenance of the workspace is required

While translations are generally very easy to implement using FME Workbench, in the longer term they can become tedious to keep up-to-date. If new feature types or attributes are added to the source data the workspace must be edited. Sometimes the data owners may not inform you their data schema has changed!

### What can be made Dynamic

There are a number of components that you could make dynamic in an FME workspace:

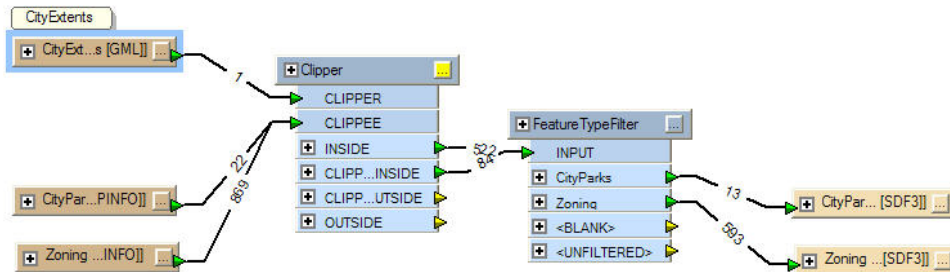
- Schema: Source Feature Types  
A workspace that would read any set of feature types from a source dataset  
*This is supported by the Merge Filter*
- Schema: Destination Feature Types  
A workspace that would read any set of feature types from a source dataset , and would write a corresponding set of feature types  
*This is supported by the Dynamic mode*
- Schema: Attributes  
A workspace that would read any set of attributes on the source feature types and write them out in the same schema  
*This is supported by Dynamic mode*
- Format  
A workspace that would write to any format, without the need to add multiple writers  
*This is supported by the Generic Writer*

## Introduction to Dynamic Workspaces

In this section you'll be introduced to dynamic workspaces. You are going to create a simple workspace to load MapInfo TAB dataset into an Autodesk SDF database. The workspace will also be able to update an existing SDF table, if new attribution is added to the source data.

You are going to update the InteropolisDatabase SDF database with a new Mapinfo TAB Zoning table and update the existing CityParks table with new Mapinfo TAB data that has an additional attribute - maintenance\_ID. You will also need to clip the datasets to match the existing Autodesk SDF data extents.

A traditional workspace to do this task would look something like this:



The disadvantage of this approach is that if a new table needs to be added to the Autodesk SDF database (destination) you have to edit the workspace. If this is part of a regular update cycle for the Autodesk SDF database you have to be careful that the data provider doesn't slip in an unexpected MapInfo TAB table, or change the schema of one of the existing MapInfo TAB tables.

In other words, workspace above is very tightly bound to the source and destination schemas.

## Dynamic Workspaces

A dynamic workspace is a workspace that is somewhat independent of the source and destination schemas.



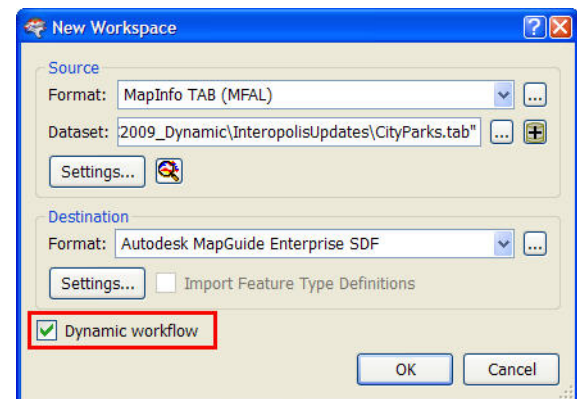
### Creating a Dynamic Workspace

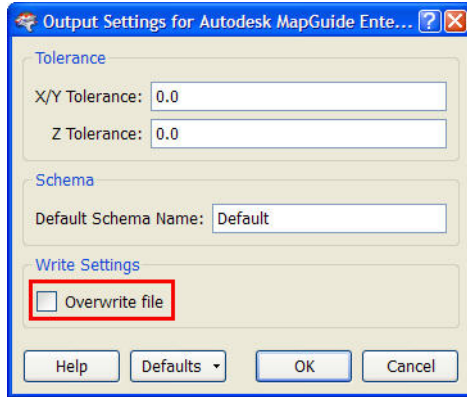
Dynamic workspaces are more easily generated when you first create a new workspace.

1. Open FME Workbench and complete the New Workspace dialog as shown on the **right**. (the data is in the directory *C:\FMEData\Resouces\UC2009\_Dynamic\InteropolisUpdates*)

Make sure you check **Dynamic Workflow**

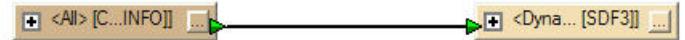
Select the **CityParks** and **Zoning** TAB files as the source dataset.





- Under the Destination Settings..., uncheck **Overwrite File** to ensure you don't delete the existing SDF database – you want to add to the database.

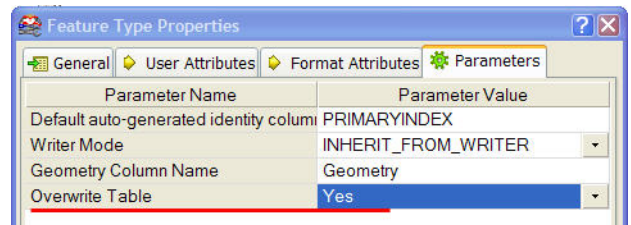
Your new workspace will have a single source feature type definition and a single destination feature type definition.



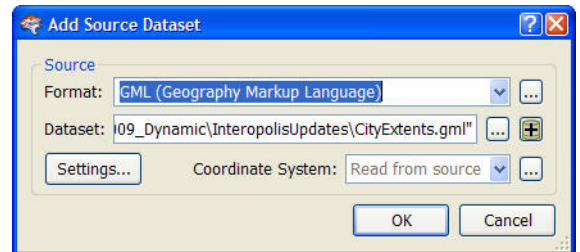
- Edit the destination feature type definition properties and under the Parameters TAB select:

**Overwrite Table: Yes.**

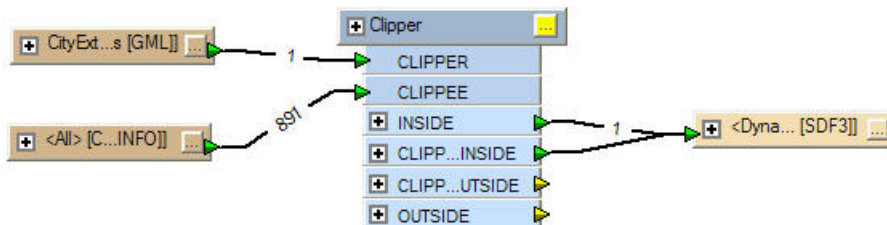
This ensures that if you update tables that already exist in the SDF database, any new attributes will be added to the SDF schema.



- Use the Source Dataset menu and Add Dataset... to add the *C:\FMEData\Resources\UC2009\_Dynamic\Interopolis Updates\CityExtents.gml* dataset – which you'll use for clipping.



- Add a **Clipper** transformer and connect it as shown below. The workspace is a bit simpler than the traditional workspace shown in the previous section.



- Run the workspace and update the *C:\FMEData\Data\DemoData\InteropolisDatabase.sdf* database. Inspect the results in FME Universal Viewer.

The Zoning table has been added to the database and the CityParks table has been updated with the maintenance\_ID attribute.

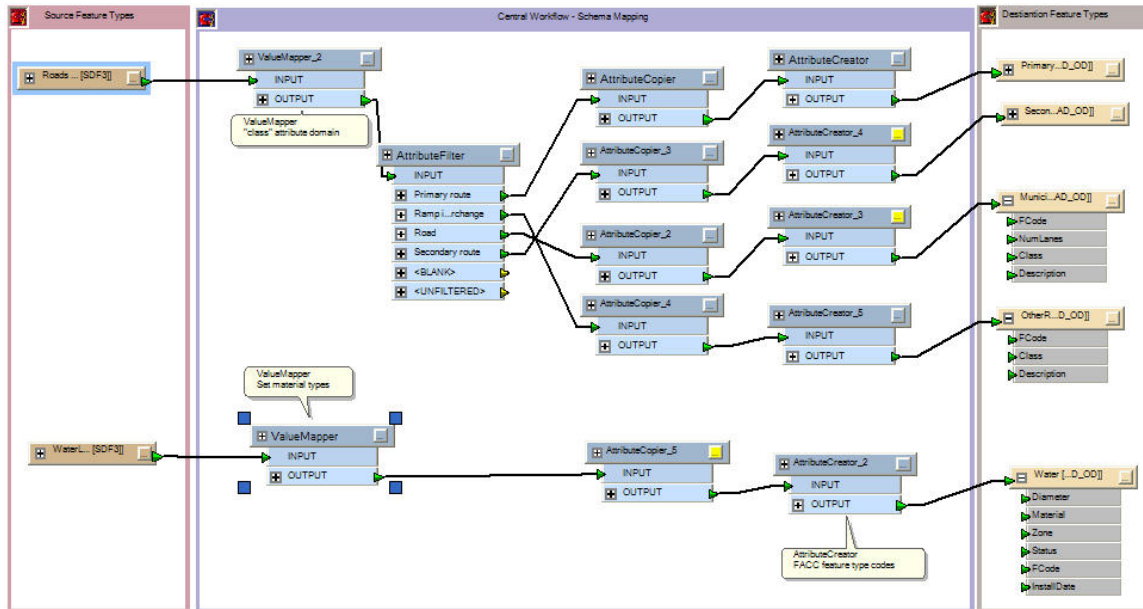
You will look at the details of how the dynamic workspace works in the next section.

## Data Distribution with Dynamic Workspaces

In this section you're going to look in more detail how a traditional workspace compares to the dynamic workspace approach. Both have their place and knowing which approach is the most suitable for a particular situation will help in the creation and maintenance of your FME projects.

### Traditional Workspaces

A common configuration of a workspace is shown below – in this example converting InteropolisDatabase.sdf database to Autodesk AutoCAD 3D Object Data:



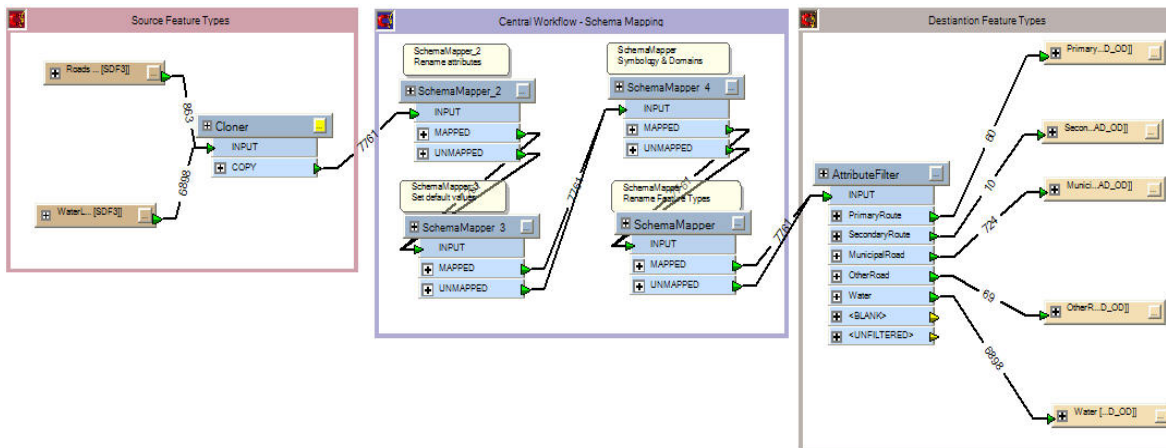
One thing you can see is the repetition of the transformers. Many of the transformers are the same, with very similar parameters. There are also some parallel pipelines – Waterlines & Roads. The other thing that stands out is that if another table is added to the source SDF database, i.e. Zoning, then there are quite a lot of steps involved in updating this workspace:

- import the source feature type definition
- add the new transformations (mostly cut & paste)
- add a new destination feature type definition
- ensure the attributes are mapped correctly (AttributeCopier)

If some of the SDF tables are updated with new attributes, there are a similar number of steps required to update the feature types.

In the long term this can require some effort to keep the workspace up-to-date with the database, or the output required by the data recipient.

The **SchemaMapper** transformer helps to reduce the repetitive nature of the workspace. This does make it somewhat easier to maintain the workspace in the long term. This is because the schema mapping is driven externally by a spreadsheet or database table, see the workspace below:



This approach to workspace design is called a “**central workflow**”. All the features pass through a common set of transformers – in this case some *SchemaMappers*. But long term maintenance of the workspace is still tedious.

To update this workspace for the Zoning and CityParks additions you’ve made to the SDF database in the first section you would have to do the following:

- update the schema mapping spreadsheet
- import or update the source feature types
- update the *AttributeFilter* with the new destination feature type names
- add new destination feature types or update existing ones.



1. Open the workspace *Example 02 Begin.fmw* and run it.
2. Inspect the results (make sure you use the Autodesk AutoCAD Map 3D Object Data reader when viewing the results – otherwise you won’t see the object data attributes).

```

-----
Features Read Summary
-----
CityParks                                     20
Roads                                         863
WaterLines                                   6898
-----
Total Features Read                           7781
-----

Features Written Summary
-----
CityParks                                     20
MunicipalRoad                               724
OtherRoad                                     69
PrimaryRoute                                 60
SecondaryRoute                               10
Water                                        6898
-----
    
```

3. Inspect the workspace and see what you would need to do to add the Zones table you just loaded in the SDF database and update the CityParks feature type.
4. If you have time, update the workspace. You can use the new schema mapping spreadsheet *SchemaMapping02.csv*.

The workspace *Example 02 Complete.fmw* shows the final result and has annotation that describes the steps needed to update the workspace. Using the *SchemaMapper* does make certain aspects of workspace maintenance easier. The next step is to create a workspace that doesn’t require any maintenance at all.

## Dynamic Data Distribution

You're going to use the dynamic workspace functionality you used in the first section to create a more versatile and easily maintained workspace than the ones above. You are going to walk through the steps of creating a dynamic workspace, the end result being a workspace that uses the schema mapping defined in the *SchemaMapping02.csv*, but has the benefits of a dynamic workspace.

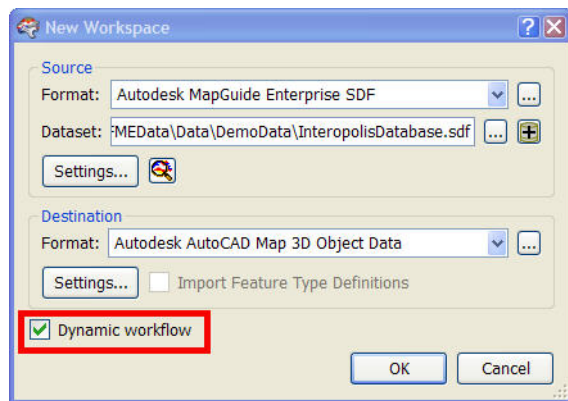


### Create a Dynamic Workspace

Rather than try to revise the workspace *Example 02 Complete.fmw*, you will start from scratch and at the appropriate time copy the *SchemaMapper* transformers from the original workspace into the new one.

1. Open a new workbench session (leave *Example 02 Complete.fmw* open).
2. Fill out the New Workspace dialog as shown on the **right**. Make sure you check the **Dynamic Workflow** check box.
3. Select the Source Format **Settings...** and check **Remove Schema Qualifier**. This removes the SDF user name from the schema.
4. Also under Settings, select the **Table List:** and choose the SDF tables to be read (all of them)

You get a very simple workspace with one source feature type definition and one destination feature type definition.



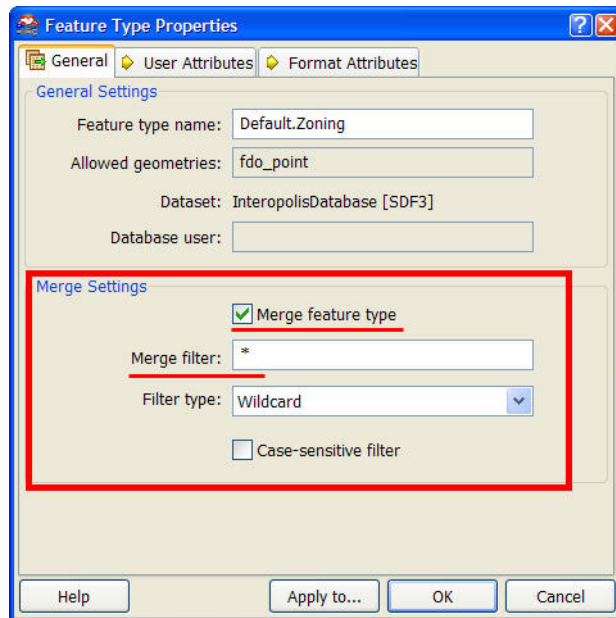
5. Run the workspace and inspect the results using FME Universal Viewer (make sure you use the Autodesk AutoCAD Map 3D Object Data reader when viewing the results – otherwise you won't see the object data attributes).

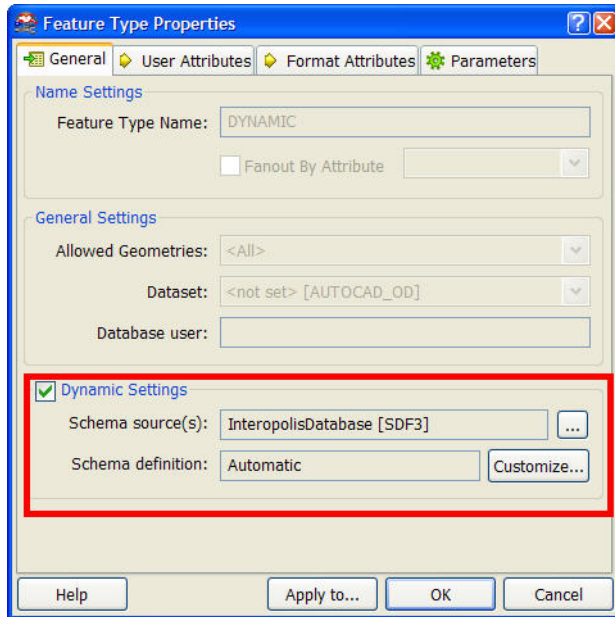
All the tables in the SDF database have been read and you have a mirror of the SDF database in the AutoCAD Map 3D output file.

### What happened under the covers?

By setting the Dynamic Workflow FME Workbench has been configured to read any table in the SDF database and the use a dynamic writer to replicate that schema for the destination dataset. This is a two step process, as described below:

6. Open the **Source Feature Type Definition** properties dialog. You can see that the **Merge Settings** has been configured. These settings allow FME Workbench to read any table in the source SDF database.





7. Open the **Destination Feature Type** definition properties dialog.

In this case the **Dynamic Settings** has been checked, enabling the dynamic mode.

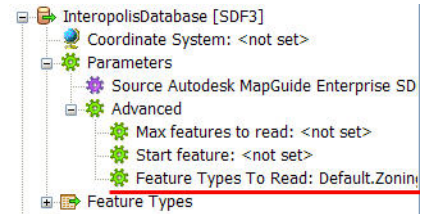
The default settings allow FME Workbench to gather the destination schema from the source feature types. When FME is run, the source dataset is scanned and the destination schema is derived from the source data. This is analogous to doing an import feature type definitions at run time.



### Choosing the source feature types

In dynamic mode, the workspace will try to read all the source data. You can use Feature Types To Read to restrict the SDF tables to be read.

1. In the workbench navigation panel, publish the **Feature Types To Read** parameter.
2. Run the workspace using Ctrl-R or **Prompt and Run Translation**



3. Select the **CityParks**, **Roads**, **WaterLines** and **Zoning** using Feature Types To Read.
4. Run the workspace. Open the resulting DWG file and inspect the results.

Only the four selected SDF tables have been translated to AutoCAD Map 3D layers.

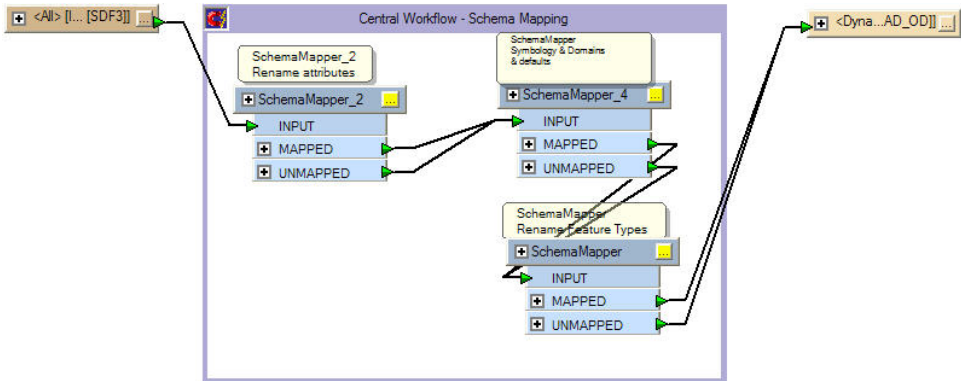


## Changing the Dynamic Destination Schema

The dynamic workspace that has been created uses the source schema to determine the destination schema – the destination schema is a mirror of the source. But you want to use the *SchemaMappers* from *Example 02 Complete.fmw* to rename feature types and attributes and set domain values.

1. Open *Example 02 Complete.fmw* and copy the contents of the **Central Workflow – Schema Mapping** bookmark into your dynamic workspace. (Use **Select All Objects In Bookmark** to copy the bookmark and its contents).

2. Disconnect the link between the source and destination feature types and route the data through the *SchemaMappers*, as shown to the **right**.



3. Run the revised workspace. You'll get an error message, something like:  
**Layer 'DYNAMIC' is not defined**

### What Happened?

By adding the *SchemaMapper*, you have renamed some of the feature types. When the dynamic writer tries to match the destination schema to the source schema, it can't find a match and generates an error.

## Schema Readers

The dynamic workspace you have created uses the source schema to determine the destination schema. But the *SchemaMapper* has been added to rename feature type and attributes and set domain values. You need to change the dynamic mode settings to derive the destination schema from somewhere other than the source schema – and avoid the error you triggered above.

Instead of copying the destination schema from the source SDF dataset, you need to configure workbench to extract the schema from a **schema template** that reflects the destination schema you want. To do this you first add a **Schema Reader** to the workspace. A Schema Reader is a special reader that doesn't read any features, but passes the schema of the feature types it scans to the dynamic writer.

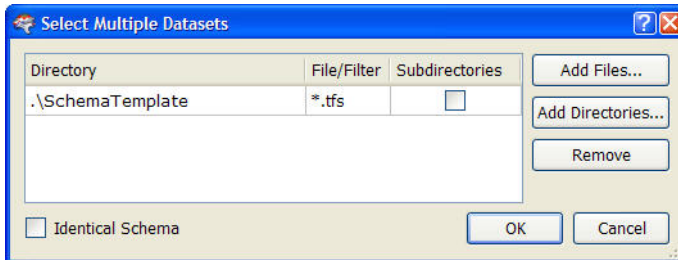
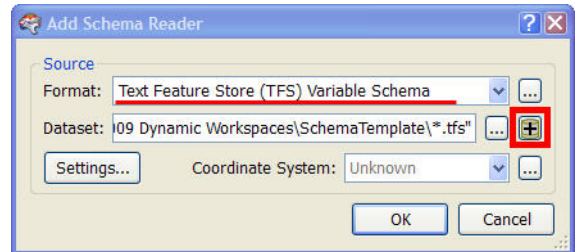
***The Schema Reader is similar to Import Feature Type Definitions. The schema can be derived from any format – it doesn't have to match either the format you're reading from or writing to. In this case you're going to use the Text Feature Store format. You're using this format because it's a simple ASCII format so you can easily inspect the schema.***



### Add a Schema Reader

1. Under the **Insert** menu select **Schema Reader...** to open the Add Schema Reader dialog (also accessible by right-clicking on the white space at the bottom of the workbench navigation pane).
2. Select the **Text Feature Store (TFS) Variable Schema** format for the schema reader.

Text Feature Store is a very simple ASCII format so it's an easy way to represent a schema template. Open one of the TFS files in a text editor to see it's content if you are interested.



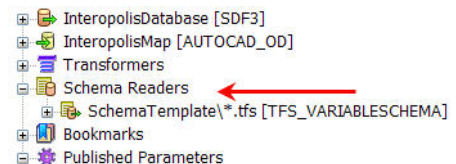
3. Use the **Advanced File Browser** to select the **.\SchemaTemplate** directory.



***If you use the simple browser, then the Schema Reader will only read the TFS schema template files you select. If another schema template is dropped into this directory the reader won't recognize it. The Advanced Browser will always read all the TFS template files.***

4. Click on **OK** on the Add Schema Reader dialog. FME will add a Schema Reader node on the navigation tree. Nothing changes on the workspace canvas.

You can now configure the dynamic writer to use the Schema Reader you've just added.

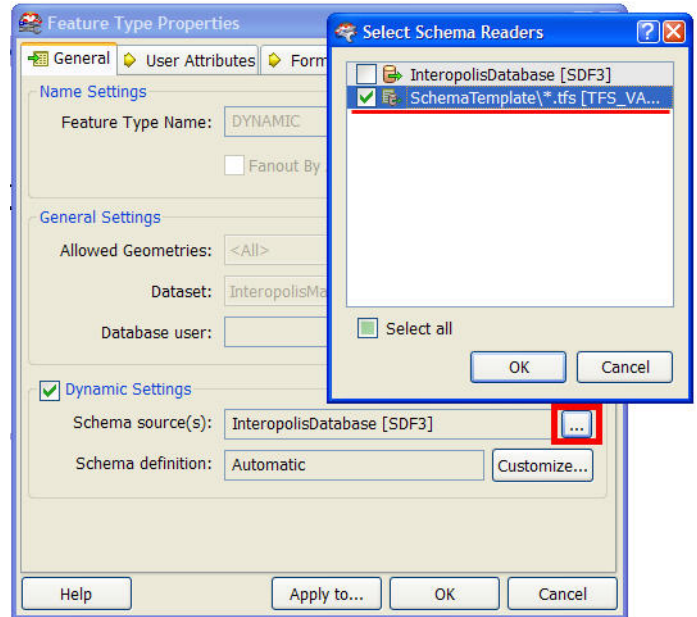




### Configure the Schema Reader

As mentioned above, the schema reader is used by the dynamic writer to obtain a schema that is different to the source dataset. You now need to configure the dynamic writer to use the schema reader.

1. Open the AutoCAD Map 3D destination **Feature Type Properties** dialog.
2. Select **Schema source(s)** to open the **Select Schema Readers** dialog.
3. Uncheck **InteropolisDatabase [SDF3]** and check **SchemaTemplate [TFS\_VARIABLE SCHEMA]**
4. Click **OK** on both the **Select Schema Readers** and **Feature Properties** Dialog.
5. Now rerun the workspace and inspect the results.



The output in the AutoCAD file now matches the mappings carried out in the *SchemaMapper* transformers.

#### **What has happened?**

The *SchemaMappers* have altered the schema of the source data, renaming feature types and attributes and mapping domain values – so now the data and the destination schema do not match. The workspace has extracted a dynamic schema using the schema reader, which matches the new schema built by the *SchemaMappers*.

The great thing is that if the schema of the SDF dataset changes, you don't have to change anything, unless you want to alter the schema of the destination data. Even then you don't need to edit the workspace. You can update the schema mapping CSVs and alter the schema template.

## Advanced Dynamic Schema Topics

There are few variations on the theme that the dynamic writer can cope with. These are feature type fanouts, more complex schema reader definitions and defining the destination feature geometry. These options are found under the Schema Definition **Customize...** button.

### Feature Type Name

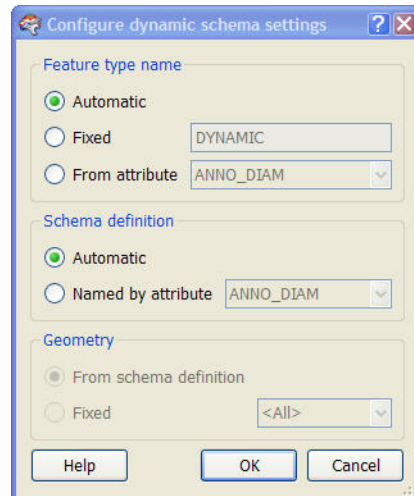
This controls the name of the destination feature type that is written. **Automatic** uses either the source schema or the schema reader feature type name. **Fixed** uses the name of the feature type in the workspace. **From attribute** has the same affect as the dataset fanout with a new feature type for each value of the specified attribute.

### Schema Definition

If the feature types in the Schema Reader do not match the `fme_feature_type`, then you can set an attribute that defines the name of the schema reader feature type to be used for the schema definitions.

### Geometry

Some formats, e.g. ESRI Shape, have a fixed geometry. This option defines where the definition for the geometry is derived from.



## Generic Writer

The Generic Writer is the tool by which a workspace can be made format independent on the destination side.

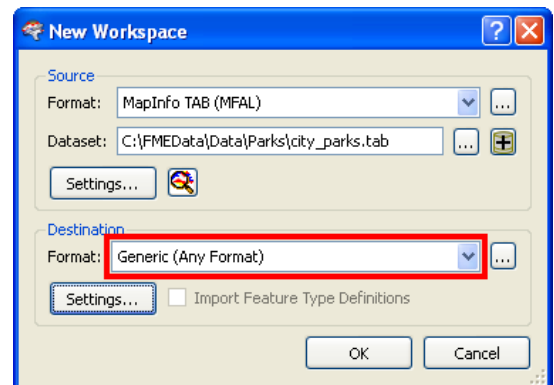
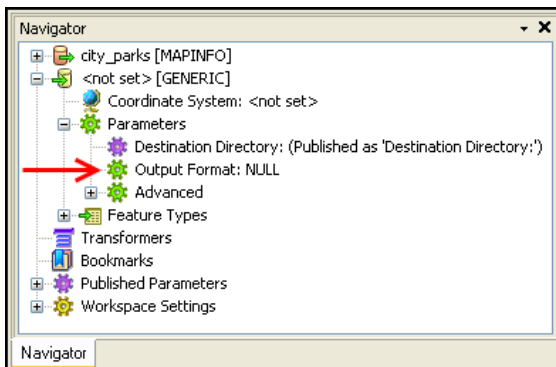
### Generic Writer Format

The generic writer is a special format that allows you to select the true destination format at the time you run the workspace.

When a workspace with generic writer is run, the destination format is determined by a parameter that can be set in the Navigation pane.

You can select the Generic (Any Format) format when you create a workspace – just like any other FME format.

Once created, the user of the generic writer workspace can choose which format to write to when the workspace is run.



## Adding the Generic Writer

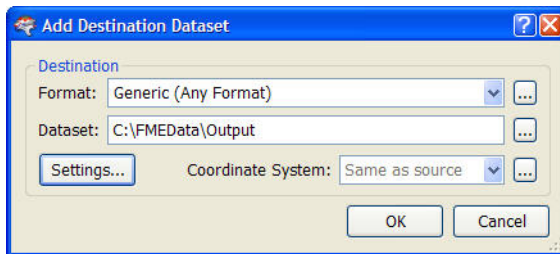
Going back to our dynamic writer workspace. You have set-up this workspace to write AutoCAD Map 3D. What happens if you get a request for traditional AutoCAD? Or someone else wants SHAPE or MapInfo TAB?



**Objective:** You wish to create a workspace which will translate the Interpolis database into a variety of formats, minimize the long term maintenance of the workspace and reduce the number of workspaces you have to maintain.

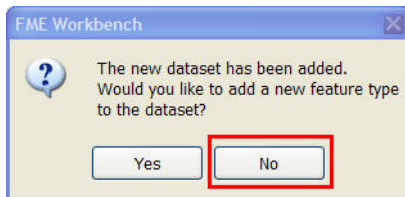
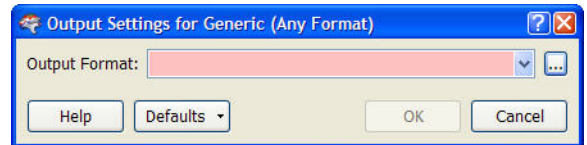
You've been asked to supply data as AutoCAD 3D Map DWG, AutoCAD DWG, ESRI Shape, Mapinfo TAB and KML. You will add a Generic (Any Format) writer to the existing workspace and then switch the feature type definition to use the new writer.

1. Continue with the Workbench from the previous section or open *Example 05 Begin.fmw*
2. From the Destination Data menu, select **Add Dataset...**



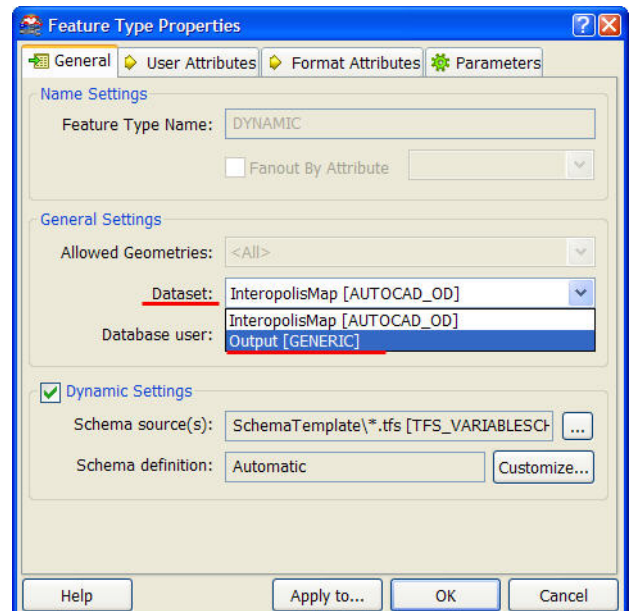
3. On the Add Destination Dataset dialog select the **Generic (Any Format)** as the format.

4. Under settings, select **Autodesk AutoCAD Map 3D Object Data** as the default Output Format.



5. At the prompt: Would you like to add a new feature type to the dataset? select **No**

6. Open the destination Feature Type Properties dialog, see image to the **right**.



7. In the General Settings panel, change the Dataset from InterpolisMap[AUTOCAD\_OD] to Output[Generic] and select **OK**
8. In the navigation panel, **delete** the InterpolisMap[AUTOCAD\_OD] dataset.

9. Use Prompt and Run Translation to start the translation.
10. In the **Edit Translation Parameters** dialog, select **Autodesk AutoCAD Map 3D Object Data** as the Output Format and click **OK**
11. Inspect the resulting DWG file in the Universal Viewer. Is it the same as our previous DWG file?
12. Rerun the workspace, but select a different output format.

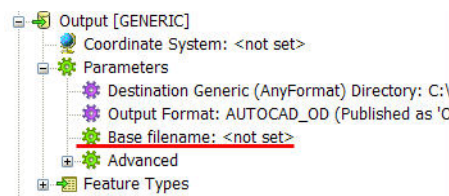
You have a workspace capable of converting the Interopolis database into pretty well any output format!

### Generic Writer Destination Dataset

By default the Generic writer uses a directory as the dataset name. This is OK if you are writing a directory based format such as ESRI Shape. If you choose a file based dataset such as AutoCAD DWG then the default behavior is for FME to reuse the directory name as the file name, i.e.:

.\output\output.dwg

You can change this behavior by setting the **Base Filename** parameter.



1. Publish the Base Filename parameter and change the name of the output AutoCAD DWG file.

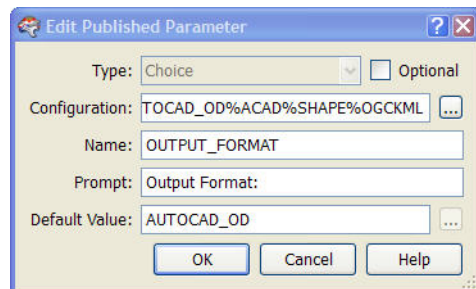
### Restricting the Generic Writer Format List

The current workspace configuration allows the user to select any format for the generic writer. It's more usual that you want to restrict the list of formats. To do this, you can add a published parameter to the workspace.



1. Continue with the current workspace or open *Example 06 Begin.fmw* for a fresh start.
2. Delete the **Output Format** published parameter.
3. In the workbench navigation pane, right click on Published Parameters and select **Add Published Parameter**.

4. Enter the parameters as shown on the *right* (you have to enter the format identifier or format short name) – **AUTOCAD\_OD, ACAD, SHAPE, OGCKML**.
5. Under the generic writer parameters, right-click on the Output Format parameter and select **Set to Published Parameter...** Select the one you just created – **OUTPUT\_FORMAT**



6. Use Prompt and Run Translation to start the translation. Select the Output Format.
7. You can reorder the published parameters by clicking on the published parameter and dragging it in the workbench navigation tree.

## Advanced Generic Writer Configuration

The generic writer is just that, a generic writer. If you find that you need to access writer specific parameters or format attributes for a specific format, then perhaps you should be using the full blown writer. However, you may need to change the default behavior of one of the formats being accessed by the generic writer.

For example, in the workspace you've been working on, you have configured the workspace to give access to the AutoCAD 3D Map, AutoCAD, Shape & KML formats. How can you use an AutoCAD Template file to define layer colors and blocks?

The generic writer can be configured to access specific writer parameters and also the feature type definition parameters for a specific writer



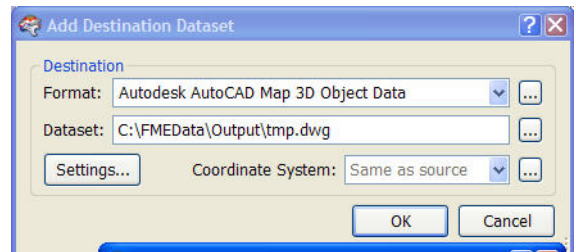
### Writer Defaults

To have the generic writer access the parameters of a specific writer you can add a dummy writer to the workspace.

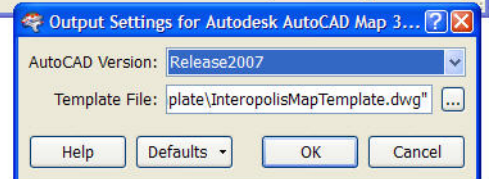
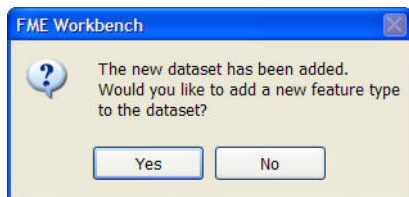
1. Continuing with the previous workspace, on the Destination Data menu, select **Add Dataset**.

Add the Autodesk AutoCAD Map 3D writer. Under **Settings**, select the *InteropolisMapTemplate.dwg* file.

You need to enter a Dataset (i.e. tmp.dwg) – even though this will never be used.



2. Click **No** at the prompt *Would you like to add a feature type to the dataset?*



When you run the workspace, and select AUTOCAD\_OD as the output format, FME will use the template file.

You can have a dummy writer for each output format the generic writer will use, if appropriate. If you add two writers, the generic writer will only use the first to obtain the parameters it needs.

3. Since the Autodesk AutoCAD Map 3D writer is never used for writing, you can **delete** the published "Destination Autodesk AutoCAD Map 3D Object Data File:" parameter.



### ***Feature Type Definition Defaults***

Some writers have parameters on the destination feature type definition and you can configure the generic writer to access these parameters as well.

1. On the Destination Data menu, select **Add Feature Type Definition**. On the General tab select the Dataset this feature type is to be associated with – in this case the AUTOCAD\_OD dataset. Change the Feature Type Name, e.g.: AutoCADParameters
2. Under the Parameters tab, change the required parameter, e.g.: Layer Frozen: yes. This parameter will be used for all layers written to AutoCAD Map 3D via the generic writer.

Note that the feature type definition is left disconnected. There are no input features, only the parameter values are used.

## Review

This session introduced the concepts of dynamic workspaces.

### **What You Should Have Learned**

The following are key points to be learned from this session:

#### ***Theory***

- **Dynamic workspaces** give maximum flexibility for minimum effort
- **Dynamic Schema** can handle source data with any schema
- **Dynamic Writer** can be used to minimize the workspace complexity
- The **Generic writer** can produce data in nearly any format

#### ***FME Skills***

- The ability to set-up a workspace in dynamic mode
- Configuration details of the Dynamic reader & writer
- The ability to use *Feature Types To Read* parameter
- How to add a Generic Writer



[www.safe.com](http://www.safe.com)

Safe Software values its customers' opinions very highly.  
To provide feedback on FME training, access the feedback form at: [www.safe.com/feedback](http://www.safe.com/feedback).  
For questions and concerns not covered by that form, please use the general feedback page at:  
[www.safe.com/company/contact/form.php](http://www.safe.com/company/contact/form.php).  
...or email the Training Manager directly at: [training@safe.com](mailto:training@safe.com).