Article

← FME Desktop (/S/Topic/0TO4Q000000QL9uWAG/Fme-...

Writing Geodatabase Relationship Classes

Jun 9, 2022 • Knowledge

Product Type
FME Desktop

FME Version
2022.0

Tutorial: Tutorial: Geodatabase Transformations (/s/article/tutorial-getting-started-with-complex-geodatabase) | **Previous:** Working with Geodatabase Feature-Linked Annotation (/s/article/working-with-feature-linked-annotation) | **Next:** Writing ArcGIS Geodatabase Attachments (https://community.safe.com/s/article/writing-arcgis-geodatabase-attachments)

Introduction

This article contains a number of writing geodatabase relationship classes examples. It includes demos for concepts covered in Introduction to Working with Geodatabase Relationship Classes (/s/article/working-with-geodatabase-relationship-classes-atta). Examples cover 1:M Relationship, Inserting into a M:N M:N or Attributed Relationship, and Advanced M:N Relationship.

For the following examples, it is assumed that the relationships have already been created in ArcGIS. This is required, as FME cannot create the relationships, but merely populate them.

***Note**: FME cannot create the relationship class in the target geodatabase. Either create the relationship class in ArcGIS/ArcPro, or use a **template (https://knowledge.safe.com/articles/40193/how-to-use-an-esri-template.html)**.*

Requirements

The Esri Geodatabase (File Geodb) reader/writer used in the following example requires that a licensed version of ArcGIS be available to the user. For more information on required ArcGIS license levels, please see Required ArcGIS License Types for FME Geodatabase Formats (https://community.safe.com/s/article/required-arcgis-license-types-for-fme-geodatabase).

Video

Content Overview

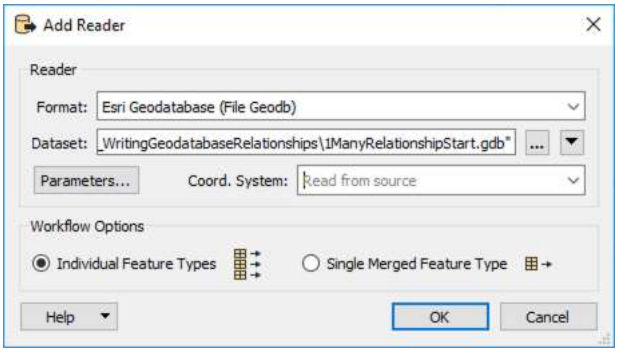
1:M Relationship Example

In this example, a 1:M relationship class translation will be demonstrated. As mentioned in the previous article, when dealing with a relationship class that is centered around a 1:M category, there is no intermediate table to keep track of these relationships. As such the relationships are bound to each other through the Primary and Foreign Keys.

Step-by-Step Instructions

1. Open FME Workbench

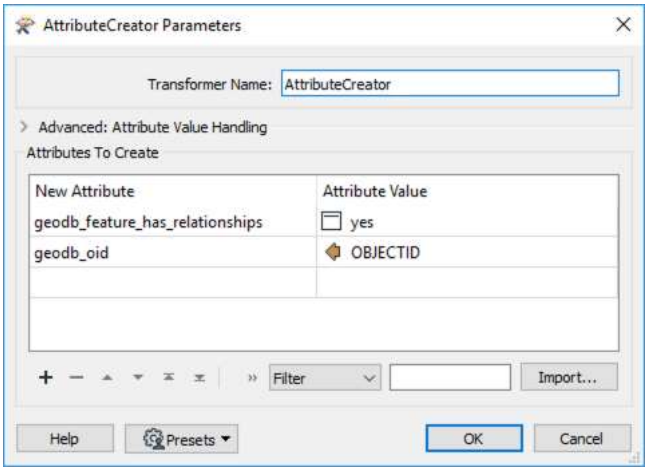
Open FME Workbench and start a blank workspace. Add an Esri Geodatabase (File Geodb) reader to the canvas and browse to the 1ManyRelationshipStart.gdb. There are no additional parameters to set for this reader, so click OK.



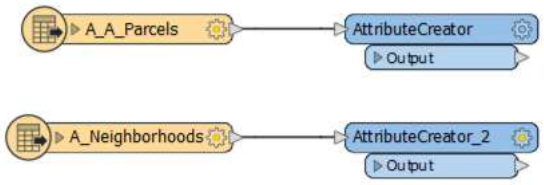
In the Select Feature Types dialog, select both feature classes (A_A_Parcels and A_Neighborhoods) and click OK.

2. Create geodb_feature_has_relationships Attribute

Add an AttributeCreator to the canvas and connect it to the A_A_Parcels reader feature type. In the parameters, create a new attribute called geodb_feature_has_relationships with the value to yes. Next, create another attribute called geodb_oid and set it to the attribute OBJECTID, then click OK.

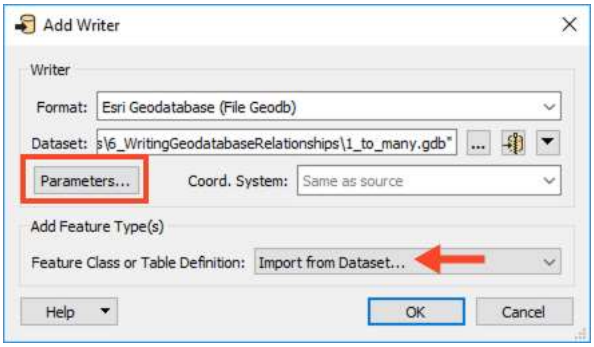


Duplicate the AttributeCreator (ctrl-d (cmd-d)) and connect it to the A_Neighborhoods reader feature type.

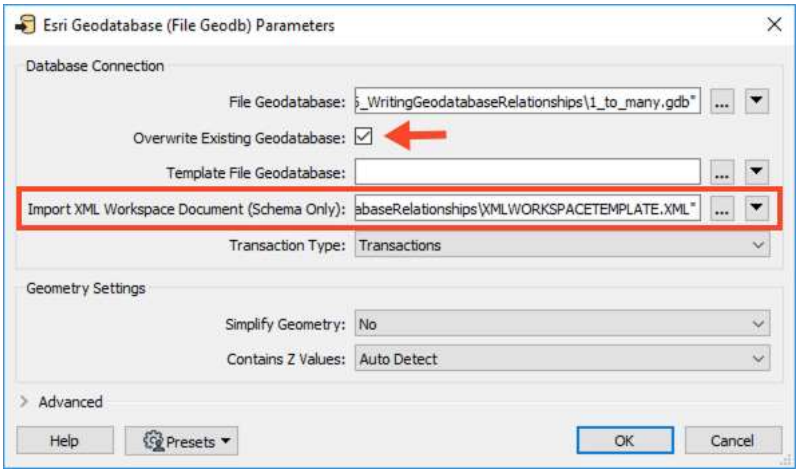


3. Write to Geodatabase

Add an Esri File Geodatabase (File Geodb) writer to the canvas and select a folder to save the output to. For this example, we named the geodatabase 1_to_many.gdb. Set the Feature Class or Table Definition to Import from Dataset, then open the Parameters.

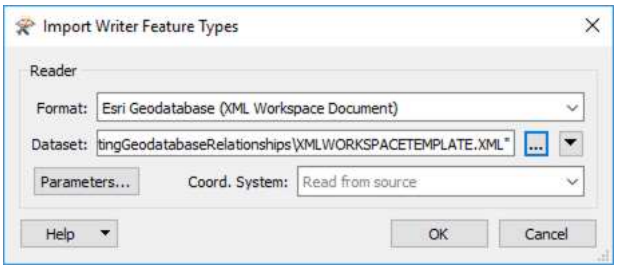


In the Parameters, enable Overwrite Existing Geodatabase, then for Import XML Workspace Document (Schema Only) browse to the XMLWORKSPACETEMPLATE.XML. Click OK twice to add the writer.



Now in the Import Writer Feature Types dialog, change the format to Esri Geodatabase (XML Workspace Document) then browse to the

XMLWORKSPAPCETEMPLATE.XML.



In the Select Feature Types dialog, select both the feature types and click OK.

Connect the writer feature types to their corresponding AttributeCreators.

4. Save and Run the Workspace.

Save the workspace, then run it. Open the geodatabase in ArcCatalog, you should be able to see which features are related. Here is an example for ParcelID: 144. It can be seen, as being related to the Kitsilano Neighborhood.

Inserting into a M:N or Attributed Relationship Example

Sometimes, the data in the origin and destination feature classes already exists and you just want to add the relationships to the relationship table. You can do this if:

- The relationship is M:N or an attributed relationship
- The relationship class already exists in your geodatabase

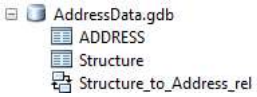
To add the relationships, set the geodb_type to geodb_relationship or geodb_attributed relationship (if attributed) and the geodb_rel_origin_oid and geodb_rel_destination_oid to their respective OBJECTID's. This does require that FME reads the feature classes which participate in the relationship but does not involve writing them out again. Only the relationship table needs to be written to in this case.

Step-by-Step Instructions

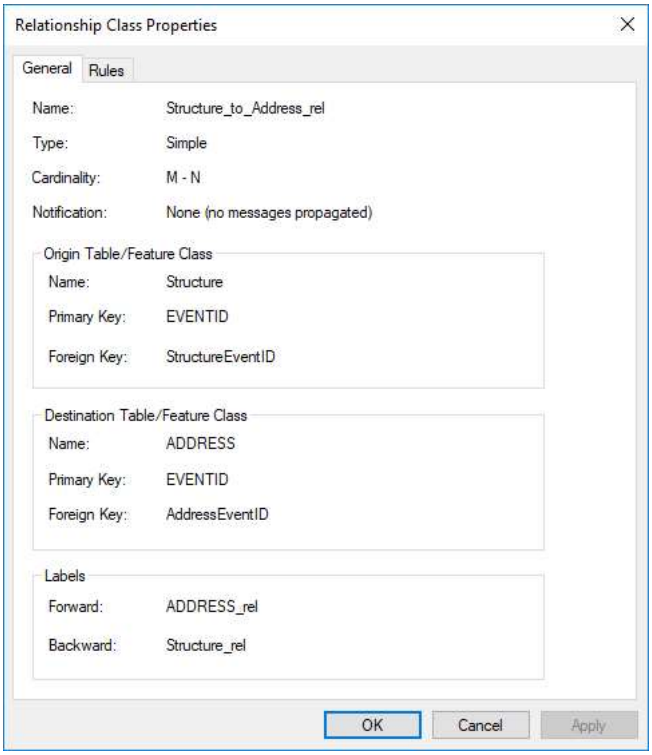
1. Review Geodatabase in ArcGIS

Before we create the relationship in FME, let's review the structure of the data in ArcCatalog. Open Esri ArcCatalog, and navigate to the Many2ManyInput.gdb file contained with the Data Download. This will be the same geodatabase that we will use in FME.

From looking at the geodatabase we can see two tables (Structure and ADDRESS) as well as a relationship class(Structure_to_Address_rel).

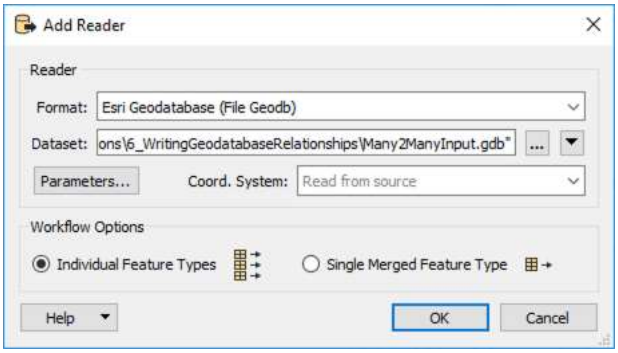


Here you can see that the relationship class has already been created. The view can be accessed through ArcGIS, and opening the properties of the relationship class. Right-click relationship class > Properties. As a reminder, this is because relationship classes must be created in ArcGIS and then they can be populated through FME.

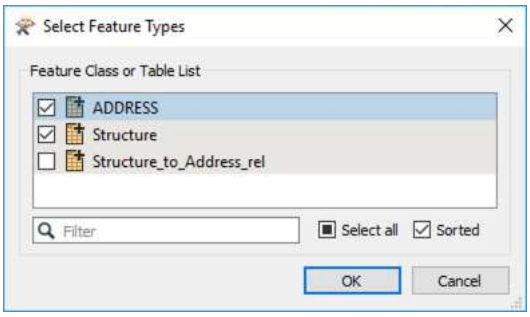


2. Add Geodatabase to FME Workbench

Now that we've inspected our geodatabase in ArcGIS and know the relationship we are working with, we can move to FME Workbench. Open a blank workspace in FME Workbench. Add an Esri Geodatabase (File Geodb) reader to the canvas and browse to the Many2ManyInput.gdb dataset, then click OK.



In the Select Feature Types dialog, select only ADDRESS and Structure, then click OK.



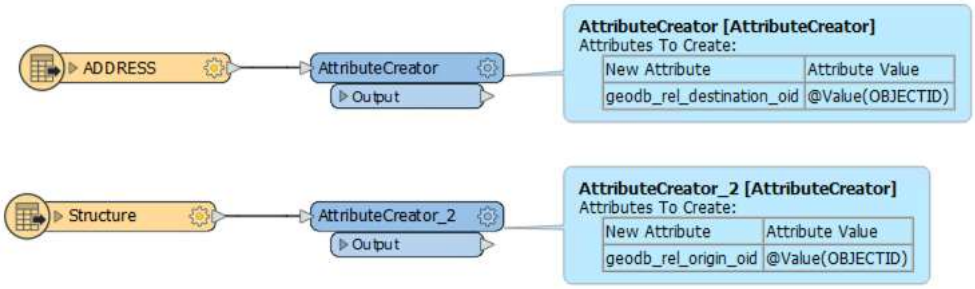
3. Modify Attributes to Reflect Relationship Class

With the fact, this is a M:N relationship we need to ensure a few attributes are stored on features to be written to the relationship class.

- geodb_rel_origin_oid - the ID attribute to be used from the related origin feature.
- geodb_rel_destination_oid - the ID attribute to be used from the related destination feature.
- geodb_type - either geodb_relationship or geodb_attributed_relationship

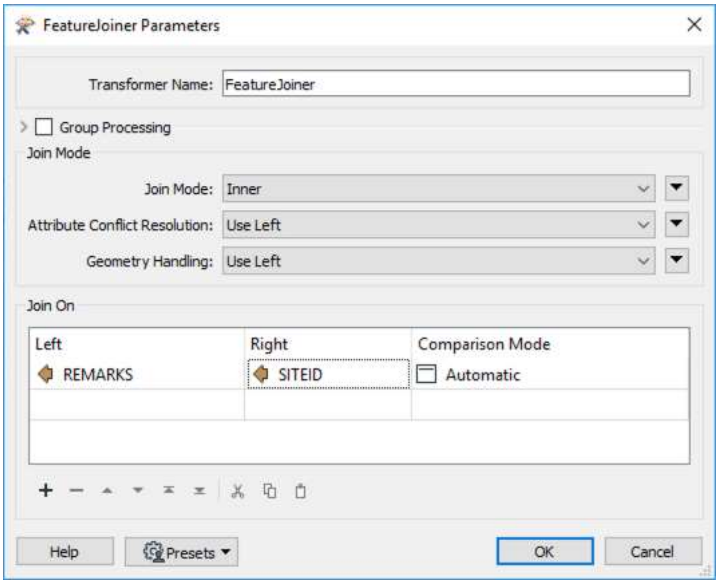
The Origin and Destination oid, refer to the primary and foreign keys on the respective feature classes. In this example, we used the ObjectID as the value for the newly created origin/destination oid. It is important to note, that these newly created values, are not explicitly written to the geodatabase, but rather help FME process how the features are related to the relationship. In order to store these features to each feature class, we need to add an AttributeCreator per feature class.

Add an AttributeCreator to the canvas and connect it to the ADDRESS reader feature type. In the parameters, create a new attribute called geodb_rel_destination_oid with the Value of OBJECTID.
Add another AttributeCreator to the canvas and connect it to the Structure reader feature type. In the parameters, create a new attribute called geodb_rel_origin_oid with the Value of OBJECTID.



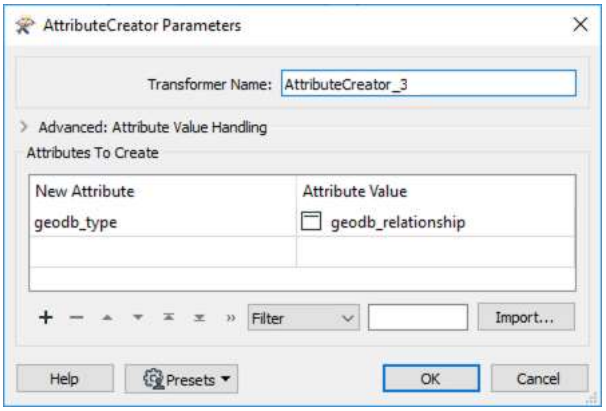
4. Merge Features

With the Primary and Foreign keys set, we can now merge the features using the FeatureJoiner. This will merge the features using the field that is shared between the two feature classes, in this case, Structure.SiteID and Address.Remarks.
Add a FeatureJoiner to the canvas and connect the ADDRESS AttributeCreator to the Left input port, and the Structure AttributeCreator_2 to the Right input port. In the parameters, set the Join On Left to REMARKS and the Right to SITEID, then click OK.



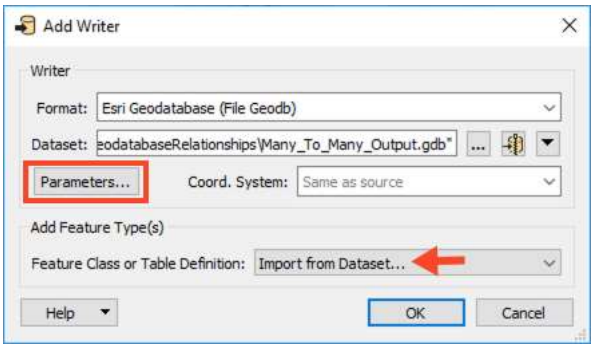
5. Set Feature Class

Lastly, another AttributeCreator transformer is required to ensure that the features are recognized as a feature class. This is done by setting the geodb_type as 'geodb_relationship', if you were dealing with an attributed relationship, the value would need to be set to 'geodb_attributed_relationship'.
Add an AttributeCreator to the canvas and connect it to the Joined output port on the FeatureJoiner. In the parameters, create a new attribute called geodb_type and set the Value to geodb_relationship, then click OK.

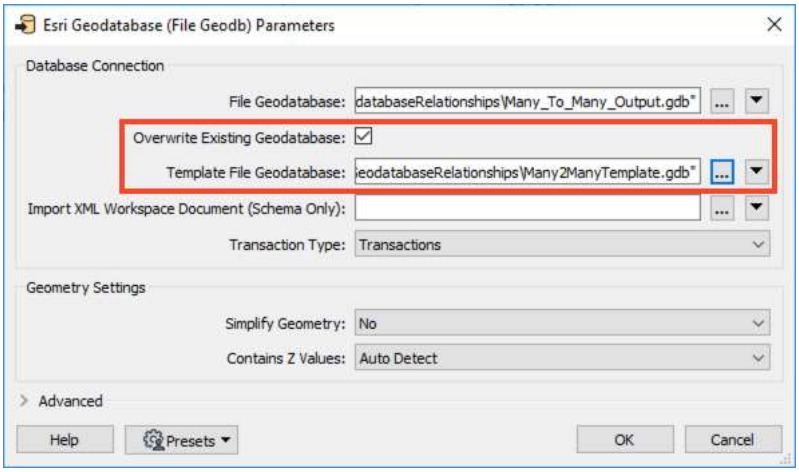


6. Write out to File Geodatabase

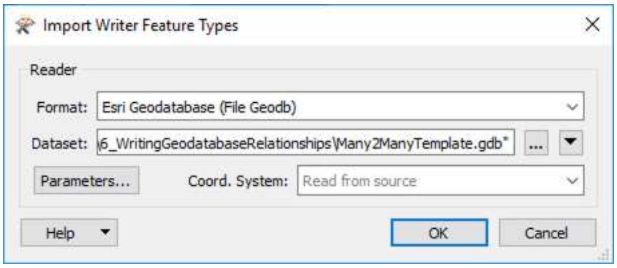
Finally, we can write out to a geodatabase. Add an Esri Geodatabase (File Geodb) writer to the canvas and browse to a location to save the geodatabase. Change the Feature Class or Table Definition to Import from Database and then open the Parameters.



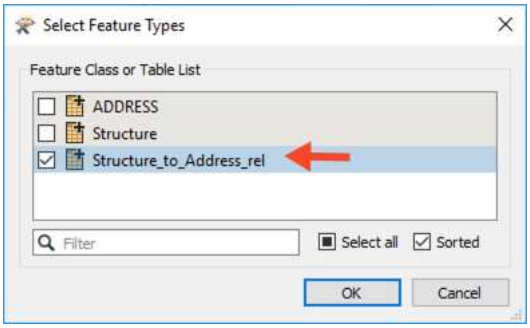
In the Parameters, enable Overwrite Existing Geodatabase and then for Template File Geodatabase browse to the Many2ManyTemplate.gbd dataset. Click OK twice to add the writer.



In the Import Writer Feature Types dialog, browse to the Many2ManyTemplate.gdb, then click OK.



Then in the Select Feature Types dialog, only select Structure_to_Address_rel, and click OK.



Once the Structure_to_Address_rel writer feature type has been added, connect it to the AttributeCreator_3.
At this stage of the workspace, we are writing directly to the relationship table. In order to do this correctly, the values we set earlier are required:

- geodb_rel_origin_oid
- geodb_rel_destination_oid
- geodb_type

We do not have to set the Foreign Key values (StructureEventID and AddressEventID) because these will be populated by the writer as long as the above values have been mapped accordingly.

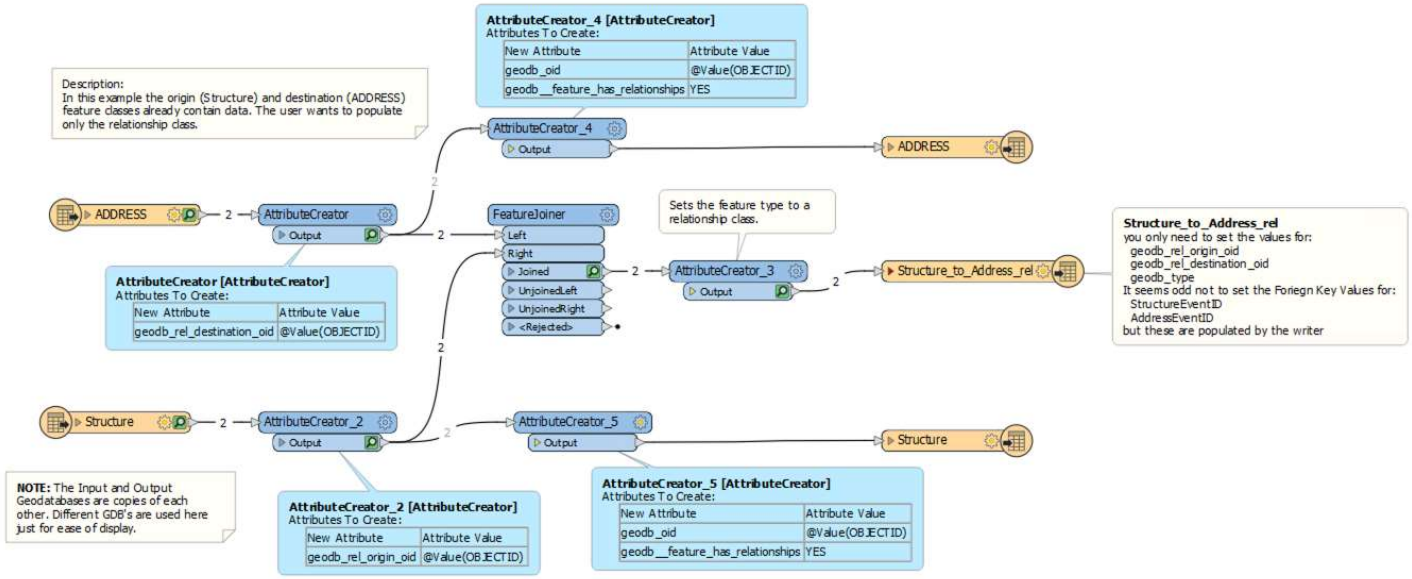
7. Save and Run the Workspace

Save and then run the workspace. The Relationship class should now be populated with the Foreign key values for AddressEventID and StructureEventId. View your output in ArcGIS.

Advanced M:N Relationship Example

In the M:N example above, the data already existed and we were just populating the relationship table to create the relation between the addresses and structures.
If you are inserting the data into a new geodatabase with a relationship class, a combination of the 1:M relationship workspace and updating a M:N relationship workspace are required to carry this task out. Again, it should be noted that the relationship class needs to be created in the destination geodatabase (or geodatabase template) prior to translation, via ArcGIS. An example of such a workspace can be seen in the 'Many2ManyAdvanced' template (see the files

section above)



Data Attribution

The data used here originates from data made available by the City of Vancouver, British Columbia (https://opendata.vancouver.ca/pages/licence/), as well as fictional data created for this article. The City of Vancouver data contains information licensed under the Open Government License - Vancouver.

Continue to the Next Article: Writing ArcGIS Geodatabase Attachments (https://community.safe.com/s/article/writing-arcgis-geodatabase-attachments)

First Published Date
7/29/2020, 12:18 AM

Last Published Date
6/9/2022, 10:29 PM

Transformation (/s/topic/0TO4Q000000...

FME Desktop (/s/topic/0TO4Q000000...

Sort by:

Latest Posts

Nothing here yet?

Log in to post to this feed.

Follow

Files (2) (/s/relatedlist/ka14Q000000wn5DQAQ/AttachedContentDocuments)

WritingGeodatabaseRelationships_2022
Jun 9, 2022 • 409KB • zip

WritingRelationshipClasses_2021.1
Jun 9, 2022 • 331KB • zip

View All
(/s/relatedlist/ka14Q000000wn5DQAQ/AttachedContentDocuments)

Related Articles

Introduction to Working with Geodatabase Relationship Classes (/s/article/working-with-geodatabase-relationship-classes-atta)

Writing ArcGIS Geodatabase Attachments (/s/article/writing-arcgis-geodatabase-attachments)

How to Read and Translate all Feature Classes from Multiple Esri Geodatabases (/s/article/how-to-read-all-feature-classes-from-multiple-esri)

Tutorial: Geodatabase Transformations (/s/article/tutorial-getting-started-with-complex-geodatabase)

Converting to Geodatabase Format (/s/article/writing-to-geodatabase-various-formats-to-geodatab)

(https://www.safe.com/legal/)

Land Acknowledgement —

Safe Software respectfully acknowledges that we live, learn and work on the traditional and unceded territories of the Kwantlen, Katzie, and Semiahmoo First Nations.