

Reading or Writing VPF using FME

The first thing to understand about VPF data is that VPF is a format, but VPF data is shipped in different "products" such as VMAP0, VMAP1, VMAP2, VMAP2i, DNC, WVS, etc. VPF products are fixed schemas, which means that for each "product" the feature classes, attributes and attribute values (including domains) are fixed. When reading VPF data FME will automatically determine the VPF product type. When writing VPF datasets using FME, you must select which VPF product you are working with. An FME Workspace created to read or write one VPF product (e.g. VMAP1), generally will not work correctly on a different product (e.g. DNC).

To a certain degree, the VPF product also determines the structure of the VPF database. VPF consists of:

```
Database (DHT file)
  Library 1 (LHT file)
    Coverage 1
      Feature Classes
    Coverage 2
      Feature Classes
    Coverage ...
      Feature Classes
  Library 2 (LHT file)
    Coverage 1
      Feature Classes
    Coverage 2
      Feature Classes
    Coverage ...
      Feature Classes
```

Depending on the dataset there may be multiple libraries, there are usually multiple coverages and coverages contain one or more feature classes (i.e roads, coastal areas, etc.). In addition, the VPF coverage may be tiled, so the feature classes may be split into 1x1 degree tiles, or some other tiling interval that also depends on the product.

FME offers two VPF readers, a database reader and a coverage reader. The FME VPF Database Reader is generally used to bulk read all the data in a VPF dataset. FME looks for the VPF DHT file and then starts to read all libraries and coverages in the database. The FME VPF Database reader is ideal for loading VPF data in a batch environment. The FME VPF Coverage reader is somewhat more selective as the workspace is configured to read individual feature classes; i.e. it's easier to pick and choose which feature classes you want to read.

Reading VPF Databases or Coverages

Reading VPF using FME is relatively straight forward, although users do sometimes require assistance or example FME Workspaces to illustrate how to make the workspaces generic so they can read any VPF dataset for the selected VPF product. You can read VPF datasets in two ways:

- VPF Coverages. Using the VPF Coverage reader you can read selected VPF feature classes by pointing FME to the *.ft files.
- VPF Databases. Using the VPF Database reader you can read an entire VPF database by selecting the DHT file. Optionally you can choose which individual feature classes you want to read.

Writing VPF Databases

VPF writing is inherently not that complex, but it is very Very VERY sensitive to the quality of the data. The VPF writer does not clean data, but it does create the required VPF topology for edges, faces and tiles. The main hurdle for writing VPF databases is that it is sensitive to the topology of the data, i.e. whether the edges of features match (or not), vertices match, whether roads spilt when they intersect and so on. So if the source data is clean and well conditioned VPF writing is likely to be successful. Feature classes in the same coverage are topologically integrated, and so these related features must be topologically clean before being sent to the VPF writer.

Tiling: Related to the topology is tiling. Some VPF products (VMAP, some DNC scales, etc.) are tiled, and in some cases the tile naming is complex, e.g. A\J\K\F. FME does create the tile names used for each different VPF product. If the VPF writer is going to tile data, then the tile coverage (i.e. tileref) **MUST** be the first coverage written to a database.

When you write to a VPF, FME will automatically tile the source data based on the appropriate tiling scheme for the product, using the tiles in the tileref coverage. FME will build the face, edge, node and cross tile topology. For this reason, your source data should NOT be tiled.

Coverages: Once the tileref has been created you can create one or more coverages. You can update a databases, but you must write an entire coverage in a single Library – individual feature classes in a coverage cannot be updated. We suggest writing one coverage at a time – first the tileref and then each coverage.

Feature Classes: VPF Feature Classes are equivalent to FME feature types. However, the FME feature type name also includes the coverage and library name, i.e. the Political Boundary Area feature class in the Boundary coverage in the “lib_078” Library will have the FME feature type name:

lib_078\bnd\polbnda

To make FME Workspace more generalized, it is usually necessary to replace the Library portion of the feature type name with a published parameter, so that the same workspace can be used to read (or write) from different libraries.

VPF Schema: Because VPF datasets are fixed schemas, with many coded domains, there may be a lot of schema and domain mapping (so the SchemaMapper transformer may help). In many cases, when writing to VPF, the source data is derived from VPF data in the first place, which means it may already closely match the VPF schema so this may not be an issue.

VPF products are usually validated against the VPF Validator application from Intergraph. This is very sensitive to topological and schema issues. Not all FME schemas have been validated against VPF Validator. If schema issues occur please contact us through www.safe.com/support, we can generally address them fairly quickly. Topology issues are harder to trace as they may be due to problems in the source data or the FME topology model. Again, please contact Safe Support and we can attempt to resolve any topology issues that may be identified in the VPF Validator.

Please contact us to find out how Safe Software can help customize a solution to meet your needs.