



Principe général :

Créer une architecture qui permette d'utiliser différents « moteurs » de lecture(/écriture) de fichiers. Ca veut dire une couche d'abstraction configurable (drivers) et qui délivre des entités de manière homogène, quelle que soit la librairie exploitée : Dtfeatures.

- Dtfeatures est une librairie générique dans DocTekus au même titre qu'une librairie pour lire/écrire des fichiers ini.
- Dtfeatures porte et délivre la structure abstraite. Un driver fait la traduction entre les objets des API et ce qui est attendu par Dtfeatures. Quelle structure d'abstraction (étendre dxf_ogr ou dxf_spatialite ? / codes dxf ? / librecad ? ...) ? Les feature class DAO d'ArcGIS sont un bon exemple.
- Dtfeatures doit potentiellement délivrer des entités CAD, mais aussi des entités SIG. Ca veut dire des géométries et des attributs. Les entités CAD embarquent en plus des propriétés CAD plus ou moins renseignées selon les capacités du driver.
- Une véritable POO semble

DocTekus – DTfeatures – Architecture théorique (ex des formats CAD)

