
mm-patch pour jump

version 0.1.2

Copyright © 2004 Michaël Michaud

Licence

Le programme mm-patch est basé sur le SIG jump [http://www.jump-project.org] publié sous licence GPL par la société vivid-solutions. Il est donc également publié sous licence GPL, et peut être utilisé ou redistribué gratuitement sous réserve que ce soit avec la même licence.

Résumé

Description de mm-patch, lot de classes modifiées de la version 1.1.2 de JUMP, permettant un typage plus précis des attributs.

Table of Contents

1. Introduction	1
1.1. Que contient mm-patch ?	1
1.2. Motivation	1
1.3. Comment installer mm-patch ?	1
2. Liste des améliorations	2
2.1. PointHandler / ShapefileHeader	2
2.2. ShapefileReader / ShapefileWriter	2
2.3. DbfFile / DbfFileWriter	2
2.4. AttributeType et AbstractBasicFeature	2
2.5. ViewSchemaPlugin	4
2.6. AttributeTablePanel	4
3. Avertissement	4
A. Historique des changements	5

1. Introduction

1.1. Que contient mm-patch ?

Vous trouverez dans ce patch :

- Des améliorations dans l'écriture et la lecture des fichiers shapefile.
- l'ajout de plusieurs types d'attributs permettant de conserver plus fidèlement la structure de vos données d'origine.

1.2. Motivation

Les changements que vous trouverez dans mm-patch ont été motivés par le souci de préserver au mieux la structure des données lors d'opérations de lecture/écriture dans des formats tels que shapefile, mif/mid, geoconcept,...

1.3. Comment installer mm-patch ?

Un petit utilitaire du nom de JarUpdater.jar est fourni pour faciliter la mise à jour de JUMP.

Placer ce fichier dans le répertoire d'installation de jump (celui qui contient jump***.jar).

Sous Windows, il suffit de double-cliquer sur JarUpdater.jar pour lancer le programme (à condition d'avoir une machine virtuelle java récente sur votre PC).

Sous d'autres systèmes d'exploitation, lancer JarUpdater.jar suivant la syntaxe habituelle ("java -jar JarUpdater.jar").

Le menu qui s'ouvre vous permet de préciser :

- Le nom du fichier jar à mettre à jour (exemple : jump-1.1.2.jar).
- le nom du fichier jar contenant les nouvelles classes (exemple : mm-patch.jar).

Valider. L'ancien fichier jar est renommé (fichier.jar.old) tandis que le nouveau fichier jar inclut les nouvelles classes.

Lancer votre application comme d'habitude.

2. Liste des améliorations

Voici résumées les principales fonctionnalités modifiées ou ajoutées par mm-patch.

2.1. PointHandler / ShapefileHeader

Correction de la classe PointHandler du paquetage org.geotools.shapefile qui ne calculait pas bien la place occupée par les différents types de géométries ponctuelles (Point, PointM, PointZ).

Dans le même ordre d'idées, ShapefileHeader a été corrigé de manière à bien écrire la plage des z dans l'en-tête d'un fichier shapefile. Cette info s'est révélée importante lors de la conversion d'un fichier shapefile dans une geodatabase, le convertisseur d'arcgis utilisant ces informations pour valider ou invalider les formes importées.

2.2. ShapefileReader / ShapefileWriter

Deux types de modifications ont été apportées à ces classes du paquetage com.vividsolutions.jump.io :

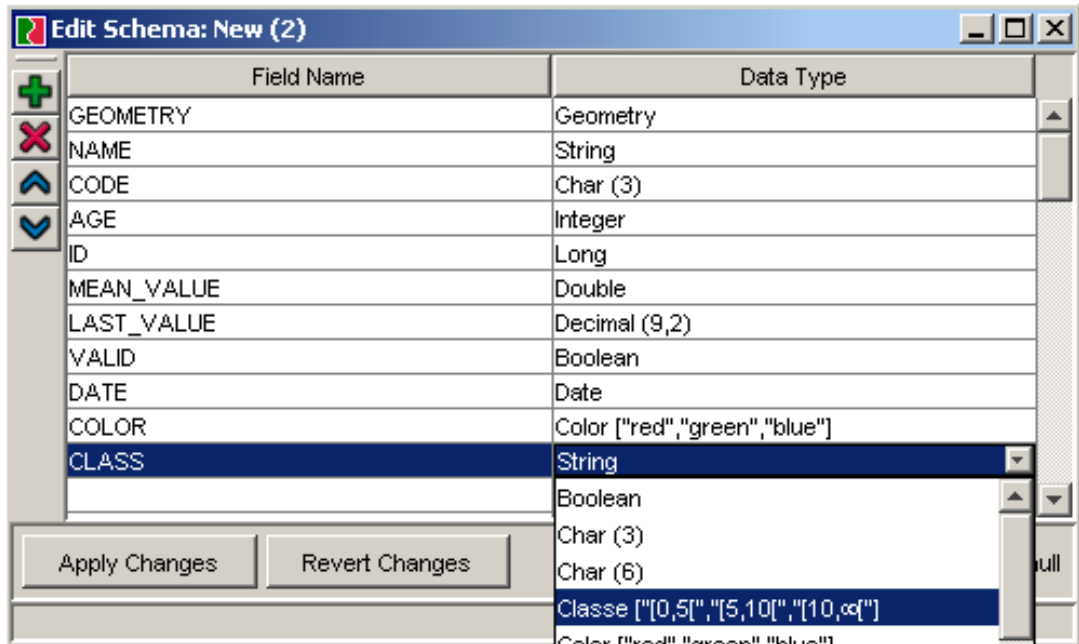
- La sauvegarde d'une couche commençant par un point ne se fait plus systématiquement dans un fichier MultiPoint. Un test préliminaire est fait. Si toutes les géométries sont des points simples, la sauvegarde est faite dans un fichier Point (resp. PointM, PointZ).
- Prise en compte des nouveaux types d'attribut booléen, décimal, char (voir ci-dessous) en lecture et en écriture.

2.3. DbfFile / DbfFileWriter

Prise en compte des nouveaux types d'attribut booléen, décimal, char (voir ci-dessous) en lecture et en écriture.

2.4. AttributeType et AbstractBasicFeature

Ajout de nouveaux types d'attributs :



- **AttributeType.BOOLEAN** (true/false) : ce type d'attribut peut être géré par les formats shapefile, mif/mid, geoconcept,...

Attention, une table ne peut être triée selon les valeurs d'une colonne de type BOOLEAN (java.lang.Boolean n'implémentant pas l'interface Comparable).

- **AttributeType.LONG** : cet attribut renforce le type INTEGER en autorisant l'écriture d'entiers compris entre -9223372036854775808 et 9223372036854775807
- **AttributeType.Char** : il s'agit d'une classe dérivée de AttributeType qui s'instancie de la manière suivante :

```
import com.vividsolutions.jump.feature.*;
AttributeType char16 =
    AttributeType.createCharAttributeType(16);
```

Ce type d'attribut est utilisé à la place de Attribute.STRING lors de la lecture des fichiers shapefile ou mif/mid.

Pour créer un nouveau type d'attribut Char, vous pouvez utiliser le menu View->BeanShell... et taper le code ci-dessus.

- **AttributeType.Decimal** : il s'agit d'une classe dérivée de AttributeType qui s'instancie de la manière suivante :

```
import com.vividsolutions.jump.feature.*;
AttributeType dec =
    AttributeType.createDecimalAttributeType(8, 2);
```

Création d'un attribut décimal (utilisation de la classe java.math.BigDecimal) de longueur totale 8 caractères et dont la partie décimale fait 2 caractères. L'interface d'import/export shapefile a été revue pour prendre en compte ce nouvel attribut.

Pour créer un nouveau type d'attribut Decimal, vous pouvez utiliser le menu View->BeanShell... et taper le code ci-dessus.

- **AttributeType.Enumeration** : le plus novateur des nouveaux attributs permet de définir une liste de valeurs possibles. La création d'un tel attribut se fait de la manière

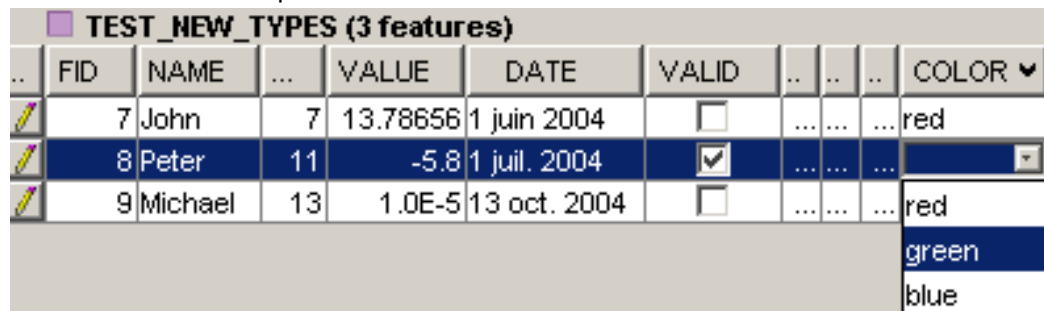
suivante :

```
import com.vividsolutions.jump.feature.*;
AttributeType color =
    AttributeType.createEnum("COLOR", new Object[]{"R", "G", "B"});
```

Pour créer un nouveau type d'attribut énuméré, vous pouvez utiliser le menu View->BeanShell... et taper le code ci-dessus en l'adaptant à vos besoins.

Une fois créé, le nouveau type d'attribut est proposé dans le menu d'édition du schéma de données, lors de la création d'un nouvel attribut (la couche doit être modifiable).

Dans une table attributaire, un attribut de type `Enumeration` peut être saisi à l'aide d'un menu déroulant. Lorsqu'il est exporté dans un format tel que shapefile, le type `Enumeration` est simplement considéré comme du caractère.



..	FID	NAME	...	VALUE	DATE	VALID	COLOR
	7	John	7	13.78656	1 juin 2004	<input type="checkbox"/>	red
	8	Peter	11	-5.8	1 juil. 2004	<input checked="" type="checkbox"/>	<div>▼</div>
	9	Michael	13	1.0E-5	13 oct. 2004	<input type="checkbox"/>	red
										green
										blue

Conversion : si vous forcez la conversion d'un champ de type caractère en type `Enumeration`, ce dernier est automatiquement complété avec les valeurs rencontrées. Ce type de comportement, qui change la définition de l'énuméré au passage, est à double tranchant. Tout commentaire sur cette option est la bienvenue.

Exemple : pour convertir un champ `STRING` qui prend les valeurs "Forêt", "Eau", "Habitat", "Autre" en `Enumeration`, créer un type `Enumeration` vide (`AttributeType.createEnum("OCCSOL", new Object[]{})`) et transtyper le champ `STRING` en champ de type `OCCSOL`. Les valeurs "Forêt", "Eau", "Habitat", "Autre" seront automatiquement ajoutées aux valeurs possibles de "OCCSOL" dans leur ordre d'apparition. L'intérêt de cette manipulation est de pouvoir remplir le champ des nouveaux objets à l'aide d'un menu déroulant (et sans se soucier de l'orthographe) plutôt que de taper les valeurs à la main.

2.5. ViewSchemaPlugin

Ce plugin du paquetage `com.vividsolutions.jump.workbench.ui.plugin` contient toute la mécanique de transtypage des attributs qui sont possibles dans l'éditeur de schéma. Il a été entièrement revu pour prendre en compte les nouveaux types.

2.6. AttributeTablePanel

Cette classe, utilisée lors de l'affichage des tables attributaires, a été mise à jour afin de prendre en compte les nouveaux types, et surtout de permettre d'afficher des listes déroulantes pour les attributs de type `Enumeration` (voir copie d'écran ci-dessus).

3. Avertissement

Attention, ce patch n'a subi que peu de tests. Il se peut que vous rencontriez des problèmes de compatibilité avec d'autres parties du logiciels (import/export dans d'autres formats,...). Aucune garantie n'accompagne ce logiciel. Vous êtes toutefois invités à faire remonter les problèmes rencontrés à l'adresse mail suivante : [<michael.michaud@free.fr>](mailto:michael.michaud@free.fr)

Les principales lacunes identifiées concernent les points suivants :

- Import / Export des nouveaux types dans les formats à base de XML
- Pas de contrainte sur le type Char empêchant de saisir des valeurs dépassant la longueur définie par le type d'attribut dans l'éditeur.

Table 1. Traitement des nouveaux types d'attributs dans différents formats : le tableau précise ce qui se passe après une double opération d'export / import

	BOOLEAN	LONG	Char	Decimal	Enumeration
JUMP JML	Informations perdues	Informations perdues	Informations perdues	Informations perdues	Informations conservées comme STRING
GML 2.0					
GML (FME)	Informations perdues	Informations perdues	Informations perdues	Informations perdues	Informations perdues
Shapefile	Informations réimportées correctement typées	Limitation de l'export à 18 caractères. Informations réimportées correctement typées	Informations réimportées correctement typées	Informations réimportées correctement typées	Informations réimportées comme STRING
Mif/Mid (expérimental)	Informations réimportées correctement typées	Informations réimportées comme INTEGER	Informations réimportées correctement typées	Informations réimportées correctement typées	Informations réimportées comme Char
GeoConcept (expérimental)	Informations réimportées correctement typées	Informations réimportées comme STRING	Informations réimportées comme STRING	Informations réimportées comme DOUBLE	Informations réimportées comme Enumeration

A. Historique des changements

- **Version 0.1.2** (23 nov. 2004)
 - Correction de bug concernant la limite entre le type int et le type long dans la classe DbfFile
- **Version 0.1.1** (04 nov. 2004)
 - Correction de la classe ShapefileHeader afin de prendre en compte la plage de z dans les shapefile 3D
- **Version 0.1** (03 nov. 2004) basée sur jump 1.1.2.
 - Réhabilitation du INTEGER (java.lang.Integer), en plus d'un type LONG afin de ne pas casser la compatibilité avec d'autres PlugIn.
 - Création de la documentation au format DocBook.

- Ajout d'un CellEditor adapté à la gestion du type Decimal dans AttributeTablePanel
- Correction de bugs dans les modules de lecture/écriture
- **Version 0.0** basée sur jump 1.1.2 (version mise à disposition des utilisateurs de jump le 24 octobre 2004)
 - Ajout des types BOOLEAN, Char, Decimal, Enumeration.
 - Transformation du type INTEGER pour qu'il utilise java.lang.Long.
 - Adaptation des classes de lecture/écriture du shapefile.
 - Adaptation du plugin ViewSchemaPlugin (transtypage).
 - Adaptation de la classe AttributeTablePanel (Ajout d'un CellEditor adapté à la gestion du type Enumeration)
 - Correction de la classe PointHandler
 - Amélioration de la classe ShapefileWriter afin de générer une classe Point plutôt que MultiPoint pour des couches ne possédant que des points.