

## TP VOTable

*Cette fiche est également consultable au format HTML sur la page <http://vizier.u-strasbg.fr/~boch/tuto/>*

*L'ensemble des fichiers nécessaires au TP est téléchargeable sous forme d'archive .tar.gz : [http://vizier.u-strasbg.fr/~boch/tuto/tp\\_votable.tar.gz](http://vizier.u-strasbg.fr/~boch/tuto/tp_votable.tar.gz)  
Décompressez le fichier en tapant `tar zxvf tp_votable.tar.gz`*

*Pour toute question relative à ce TP : [boch@astro.u-strasbg.fr](mailto:boch@astro.u-strasbg.fr)*

*Ce TP a pour objectif de présenter différents outils permettant de lire, créer ou modifier des documents VOTable.*

*Un bref descriptif des différents logiciels utilisés durant ce TP est disponible sur la page <http://vizier.u-strasbg.fr/~boch/tuto/outils.htm>*

*Remarque : pour l'ensemble des exercices faisant appel à la programmation, les endroits dans le code source où effectuer les modifications sont signalés par des commentaires **A COMPLETER**.*

## 1 Conversion de tables ASCII en VOTable

Dans cette partie, nous allons voir différentes méthodes permettant de convertir des tables ASCII en VOTable afin de les utiliser dans des outils VO.

Dans un premier temps, allez sur la page VizieR : <http://vizier.u-strasbg.fr/cgi-bin/VizieR>. Entrez 2MASS dans le premier champ et validez. Choisissez la table II/246/out, validez. Entrez comme position `oph s1`, choisissez un rayon de 0.6 degrés et validez. Apparaît alors la liste des 50 premières sources répondant à votre requête. Changez le nombre maximum de réponses à 9999, et choisissez `Tab-Separated-Values` comme format de sortie. Sauvegardez le résultat sous le nom `2mass-ophs1.tsv` par exemple.

Vous pouvez ouvrir le fichier dans un éditeur de texte pour visualiser.

*Remarque : notez que VizieR propose une sortie VOTable parmi la liste des formats de sortie disponibles.*

## 1.1 Utilisation de conVOT pour convertir notre fichier en VOTable

Lancez conVOT en tapant `conVOT` dans un terminal. Choisissez `Convert ASCII`, et chargez le fichier précédemment sauvegardé.

conVOT essaye de deviner le nombre de lignes à ignorer. Vérifiez, modifiez si besoin est, et cliquez sur `OK`.

conVOT vous demande maintenant quelles sont les lignes décrivant les champs, les unités, et quelle est la première ligne de données. A nouveau, vérifiez, modifiez si nécessaire et validez. L'écran suivant demande confirmation de vos choix.

conVOT vous demande alors de valider son choix de délimiteur entre les colonnes. Vous avez ensuite la possibilité d'éditer les méta données. Nous nous contenterons d'ajouter les UCD pour chaque colonne d'après la table de correspondance suivante :

Colonne	UCD
_RAJ2000	pos.eq.ra
_DEJ2000	pos.eq.dec
RAJ2000	pos.eq.ra;meta.main
DEJ2000	pos.eq.dec;meta.main
Jmag	phot.mag;em.IR.J
e_Jmag	stat.error;phot.mag;em.IR.J
Hmag	phot.mag;em.IR.H
e_Hmag	stat.error;phot.mag;em.IR.H
Kmag	phot.mag;em.IR.K
e_Kmag	stat.error;phot.mag;em.IR.K
Qflg	meta.code.qual
Rflg	meta.code
Bflg	meta.code
Cflg	meta.code
Xflg	meta.code
Aflg	meta.code

Validez, puis validez à nouveau.

Vous pouvez alors visualiser les premières lignes du document VOTable créé. Sauvegardez le (sous le nom `2mass-ohs1.xml` par exemple). Quittez conVOT.

Lancez VOPlot (commande `voplot`), et vérifiez que vous pouvez y charger le fichier nouvellement créé.

## 1.2 Conversion d'une table FITS ASCII en VOTable via top-cat

Pour cette partie, nous disposons d'une table FITS ASCII issue d'un catalogue de modélisation d'évolution des populations stellaires (sans coordonnées). Il s'agit d'un échantillon de la table globale, seules les étoiles de métallicité 0.0002 et d'âge compris entre  $10^8$  et  $10^9$  années ayant été retenues. Le fichier se nomme `evolution.model-sample.fit` et se trouve dans le dossier `~/votable`.

Lancez topcat par la commande `topcat`. Sélectionnez `File --> Load New Table` et chargez le fichier en question.

Sélectionnez `Table Views --> Table Data`. Les données sont affichées dans une vue semblable à celle d'un tableur : vous pouvez trier par colonne, modifier leur emplacement, changer la valeur des cellules, etc.

Refermez cette fenêtre. Sélectionnez `File--> Save Table`. Choisissez `votable-tabledata` comme format et sauvegardez sous le nom `evolution_model-sample.xml` par exemple. Quittez topcat

*NB : topcat inclut un utilitaire de conversion de tables numériques en ligne de commande : **tablecopy** qui s'avère pratique pour automatiser la conversion de nombreuses tables. Tapez `tablecopy -h` sur la ligne de commande pour afficher les nombreux formats de sortie supportés par tablecopy.*

Lancez maintenant VOPlot (commande `voplot`). Sélectionnez `File --> Open`, et chargez le fichier VOTable précédemment créé. Nous allons pouvoir tracer un diagramme HR. Choisissez la colonne `logL` (luminosité) pour ordonnée, et la colonne `logTe` (température) pour abscisse. Appuyez sur `Plot` pour effectuer le tracé. Inversez l'axe des abscisses (case à cocher `Rev`). On distingue la séquence principale d'un diagramme HR.

Vous pouvez également tracer `logL` en fonction de `logT` pour apprécier l'évolution de la luminosité des étoiles selon leur âge (sélectionnez le menu `Mode` puis `Zoom mode` pour effectuer un zoom dans une partie du graphe).

Quittez VOPlot.

### 1.3 Conversion tables ASCII vers VOTable "à la main"

Nous allons voir rapidement 2 exemples de programme permettant de convertir des fichiers TSV en fichier VOTable.

#### 1.3.1 Perl

Aucune librairie spécifique n'est nécessaire pour cet exemple.

Exécutez `tsv_to_vot.pl` (dans le répertoire `~/votable/perl`) avec comme argument un fichier TSV. En l'état actuel, le script produit un VOTable incomplet. Ouvrez le script dans votre éditeur de texte, et complétez les 2 lignes signalées par **A COMPLETER**.

Relancez le script sur un fichier TSV. Redirigez `STDOUT` vers le fichier `test.xml`. Lancez la commande `xmllint --noout --valid test.xml` afin de tester la validité du fichier produit par rapport à la DTD VOTable.

*NB : La "correction" se trouve dans le fichier `tsv_to_votable-corrected.pl`*

#### 1.3.2 Java

On utilisera le module "writer" du parser SAVOT pour cet exemple. Il s'agit de construire en mémoire la structure du document VOTable en suivant le modèle interne de SAVOT.

Le projet se trouve dans `~/workspace/TSVToVOTable/`. Vous pouvez utiliser Eclipse ou votre éditeur de texte favori. Vous pouvez compiler en tapant `ant` à la

racine du projet. Pour lancer l'application : `./run.sh INPUT_FILE [OUTPUT_FILE]`  
La Javadoc pour le modèle interne de SAVOT se trouve sur  
<http://cdsweb.u-strasbg.fr/cdsdevcorner/savot2.1.3/doc1/index.html> , la Javadoc pour  
la partie "writer" se trouve sur  
<http://cdsweb.u-strasbg.fr/cdsdevcorner/savot2.0/doc1/cds/savot/writer/WriteDocument.html>

La correction se trouve dans `~/workspace/TSVToVOTable-corrected`.

*NB : On pourrait bien entendu rendre ces exemples plus complexes en ajoutant les UCDs par le biais d'une table de correspondance, en acceptant plusieurs délimiteurs de colonnes, etc...*

## 2 Parser des documents VOTable

Nous verrons dans cette partie comment lire des documents VOTable via un parser.

### 2.1 Perl : utilisation du parser Astro::VO::VOTable

Dans le répertoire `~/votable/perl`, exécutez le script `print_votable.pl` sur le fichier `eis_all_tables.xml`

Ce script lit un document VOTable et l'affiche dans un format lisible. Editez le script et complétez-le pour afficher non seulement le nom de chaque champ (FIELD) mais aussi son UCD et son unité (utilisez les méthodes `get_ucd` et `get_unit`) Vous pourrez utiliser le fichier `eis_all_tables.xml` comme entrée.

Le script corrigé se nomme `print_votable-corrected.pl`

### 2.2 Utilisation de SAVOT

Le projet se trouve dans `~/workspace/PrintVOTable`. Comme précédemment, tapez simplement `ant` pour compiler.

Pour exécuter le projet : `./run.sh VOTABLE_FILE`.

En vous aidant de la Javadoc de SAVOT

(<http://cdsweb.u-strasbg.fr/cdsdevcorner/savot2.1.3/doc1/index.html>) , complétez la classe pour afficher l'ensemble des champs (FIELD) d'une table, et pour afficher les données.

A nouveau, testez l'application sur le fichier `eis_all_tables.xml`

Le projet corrigé se trouve dans `~/workspace/PrintVOTable-corrected`

## 3 Edition/Modification de documents VOTable

### 3.1 En utilisant topcat : diagramme couleur-couleur

Pour cet exemple, nous allons reprendre le fichier `2mass-ohs1.xml` produit précédemment (si vous ne l'avez plus, il est disponible dans le répertoire `~/votable/`).

Lancer topcat (commande `topcat`), et ouvrez le fichier `2mass-ohs1.xml`. Allez dans **Table Views** --> **Columns Info**. Nous allons ajouter 2 colonnes à ce document afin de pouvoir tracer un diagramme couleur-couleur. Cliquez sur le "+" vert. Entrez un nom pour la nouvelle colonne (J-H par exemple). Puis, entrez dans

**Expression :**  $Jmag - Hmag$ . Cliquez sur OK. Effectuez la même opération pour ajouter la colonne H-K.  
Fermez la fenêtre d'information sur les colonnes, et sélectionnez **Table views --> Table Data**. Déplacez l'ascenseur horizontal pour apercevoir les nouvelles colonnes créées.  
Sauvegardez le fichier avec les 2 colonnes supplémentaires (sous le nom `2mass-ohs1_new_cols.xml` par exemple).

*NB : VOPlot permet également d'effectuer ce genre d'opération sur les colonnes. Un des avantages de topcat est qu'il supporte l'ensemble des fonctions de java.Math*

### 3.2 En utilisant VOPlot: diagramme couleur-couleur

Lancez Aladin, et chargez-y le fichier contenant les nouvelles colonnes (**LOAD --> MyData**). Sélectionnez un ensemble de sources, et notez dans la fenêtre des mesures (en bas) la présence des méta-données (unités, UCD).

Interrogez le serveur d'images Aladin (**LOAD--> Aladin**) et sélectionnez l'image SERC de taille 1.7'x1.7'.

Lancez VOPlot (depuis le menu Tools).

Dans VOPlot, tracez J-H en fonction de H-K. Remarquez que les sources sont relativement dispersées. On va choisir un sous-ensemble en conservant uniquement les sources dont l'erreur sur la magnitude est petite. Sélectionnez le bouton **New Filter**, entrez  $eK < 0.1$  comme description du filtre puis entrez  $\$7 < 0.1$  comme expression. Validez (OK).

Sélectionnez le filtre nouvellement créé, et cliquez sur **Plot**. Cochez la case à cocher **Overlay**, sélectionnez le filtre **All** et validez à nouveau.

Affichez maintenant uniquement le filtre  $eK < 0.1$ , et sélectionnez les objets présentant un excès en K. Les objets correspondant sont également sélectionnés dans la fenêtre Aladin : ce sont des objets jeunes qui se trouvent dans la zone d'absorption.

### 3.3 En Java : ajout d'une colonne flux à partir de la magnitude

La conversion de flux en magnitude est donnée par la formule suivante :

$$F = 10^{0.4*(Z_p - M)}$$

où F est le flux en Jy,  $Z_p$  le point-zéro du système photométrique utilisé, et M la magnitude.

Nous allons construire un petit programme qui va modifier un document VOTable pour ajouter une colonne **flux** pour une magnitude donnée.

Pour notre exercice, nous utiliserons le fichier `eis_I_band.xml`, extrait de la table *bande I* du catalogue EIS (ESO Imaging Survey). Pour cette bande, le valeur du point-zéro est de 8.4577. Le nom du champ magnitude est **Mtot**

En utilisant SAVOT, le principe pour modifier un document VOTable est le suivant :

- chargement du document en mémoire dans un objet SavotVOTable

- modification de cet objet
- sérialisation de l'objet dans un flux de sortie

Le projet se trouve dans `~/workspace/FluxConversion`. Complétez les lignes signalées par **A COMPLETER**.  
Compilez en lançant `ant`, et exécutez le programme en tapant `./run.sh eis-I_band.xml`.

Le projet corrigé se trouve dans `~/workspace/FluxConversion-corrected`