
jump-sqi

Système de requêtes pour JUMP

Copyright © 2004 Michaël Michaud

Licence

Le plugin jump-sqi, ou "jump simple query interface" est conçu pour fonctionner avec le SIG jump [<http://www.jump-project.org>] publié sous licence GPL par la société vividsolutions. Le plugin est également couvert par la licence GPL. Il peut donc être utilisé ou redistribué librement et gratuitement, et n'est accompagné d'AUCUNE GARANTIE .

Version 0.1.0
Historique des versions
09 déc 2004

Résumé

Ce document décrit les fonctionnalités et la manière d'utiliser jump-sqi, un petit composeur de requête pour JUMP.

Table des matières

1. Introduction	1
1.1. Qu'est ce que jump-sqi ?	1
1.2. Que puis-je faire avec jump-sqi ?	1
1.3. Limites de jump-sqi ?	2
1.4. Installation	3
2. Configurer l'interface de requête	3
2.1. Le gestionnaire de requête	3
2.2. Filtrer les attributs	3
2.3. Définir la présentation du résultat	4
2.4. Le composeur de requête	4
2.5. Lancer la requête	9
A. Historique des changements	9

1. Introduction

1.1. Qu'est ce que jump-sqi ?

Jump-sqi est un petit composeur de requête adapté à l'environnement JUMP. Une interface graphique simple et conviviale vous permet d'interroger les couches de données affichées dans la tâche active, et de renvoyer le résultat sous la forme d'une nouvelle couche, ou simplement de sélectionner les objets trouvés. Il permet d'effectuer :

- Des requêtes spatiales.
- Des requêtes sur attribut

1.2. Que puis-je faire avec jump-sqi ?

1.2.1. Contexte

L'environnement de l'interface de requête possède les caractéristiques suivantes :

- possibilité d'enregistrer / de rappeler une requête
- la composition des requêtes est assistée par de nombreux menus déroulant
- possibilité de filtrer le types de requêtes possibles en fonction du type d'attribut (caractère, numérique ou géométrique)
- possibilité de rendre les requêtes de type caractère insensibles à la casse

1.2.2. Requêtes

Le système de requête de jump-sqi permet :

- d'effectuer des requêtes à partir d'une ou plusieurs couches
- d'effectuer des requêtes portant sur tous les types d'attribut (sauf Object)
- d'effectuer des requêtes spatiales (intersecte, contient, est inclu, touche,...)
- d'appliquer des fonctions de traitement de chaînes de caractères aux attributs de ce type (extraction d'une sous-chaîne, retrait des blancs initiaux et finaux,...)
- d'appliquer des fonctions géométriques (ex. buffer, centroïde, longueur, surface...)
- de paramétrer certains types de fonctions ou d'opérateurs (sous-chaîne de caractères, distance)
- d'utiliser les expressions régulières sur les chaînes de caractères (opérateurs match et contient)

1.2.3. Résultats

Les résultats de la requête peuvent être matérialisés de trois façons différentes :

- sélection des objets trouvés dans la fenêtre d'affichage
- ouverture d'une table attributaire contenant le résultat
- création d'une nouvelle couche contenant le résultat

1.3. Limites de jump-sqi ?

Les limitations de jump-sqi sont les suivantes :

- Les objets sélectionnés le sont entièrement et avec tous leurs attributs (aucune possibilité de définir un schéma de donnée cible particulier)
- il n'est pas possible de définir des requêtes sur des critères multiples avec des "et" et des "ou" (le "et" peut facilement être simulé en effectuant une requête sur le résultat de la requête précédente).
- il n'est pas possible de faire de jointure

- il n'est pas possible d'effectuer de requête de type SQL

1.4. Installation

1.4.1. Installation du plugin pour une utilisation sous jump

Il vous suffit pour celà de placer les deux fichiers jar (jump-sqi.jar et buoy.jar) dans le répertoire de jump dédié aux plugins (répertoire ext par défaut).

1.4.2. Développement

Pour améliorer jump-sqi ou l'adapter à vos besoins, utilisez le source situé dans le même fichier jar que l'exécutable (jump-sqi). Vous aurez besoin pour la compilation des bibliothèques jump, jts, et buoy.

1.4.3. Internationalisation

A la racine du fichier jar, se trouvent les fichiers QueryDialog_i18n_ll_CC.properties nécessaires à la traduction de l'interface. Si le fichier correspondant à votre langue ne figure pas, prenez le fichier par défaut (QueryDialog_i18n.properties), copiez-le, complétez-le et renommez-le en y ajoutant les deux lettres correspondant à votre langage et les deux lettres correspondant à votre pays.

2. Configurer l'interface de requête

La moitié supérieure du panneau de requête permet de configurer l'environnement de la requête.

2.1. Le gestionnaire de requête

Le gestionnaire de requête contient deux boutons, Ouvrir et Enregistrer sous..., permettant d'ouvrir une requête existante ou de sauvegarder la requête courante.

A l'ouverture d'une requête préenregistrée, un message d'erreur peut s'afficher dans les cas suivants :

- la couche sur laquelle doit être effectuée la requête n'est pas chargée
- l'attribut sur lequel est défini la condition n'existe pas dans la couche interrogée



Les requêtes sont enregistrées dans un fichier de propriété (fichier texte définissant des paires clé/valeur sous forme de chaînes de caractères. Attention toutefois, ces fichiers ne sont pas fait pour être édités à la main, et le paramétrage d'une requête avec un éditeur de texte peut s'avérer délicat.

2.2. Filtrer les attributs

Le deuxième sous-menu permet de filtrer les attributs suivant 3 types :

- caractères
- numérique

- géométrique

Seuls les attributs correspondant aux types cochés dans ces cases sont proposés dans le constructeur de requête (le panneau situé au-dessous).

De plus, le fait de cocher la case caractères fait apparaître une case à cocher supplémentaire qui définit si les comparaisons de chaînes de caractères doivent être ou non sensibles à la casse :

Panneau intitulé "Filtrer les attributs de type" :

- ☒ Caractère
- ☒ Numérique
- ☒ Géométrique
- ☐ Sensible à la casse

Dans le cas présenté ci-dessus, tous les attributs seront proposés, et les comparaisons de chaînes de caractères seront insensibles à la casse :

Une requête portant sur les objets ayant pour nom lyon trouvera aussi bien lyon, que LYON, Lyon, LyOn,...

2.3. Définir la présentation du résultat

Le résultat d'une requête peut être définie de trois façons différentes qui peuvent se combiner entre elles.

Panneau intitulé "Résultats" :

- ☐ Afficher la table
- ☒ Sélectionner
- ☐ Créer une nouvelle couche

1. Afficher la table : le résultat de la requête apparaîtra sous la forme d'une table attributaire avec tous les objets de la ou des couches définies, et correspondant au critère de sélection défini.
2. Sélectionner : les objets trouvés sont sélectionnés dans la carte. Ils peuvent appartenir à plusieurs couches différentes. Dans la mesure où le compositeur de requêtes permet de prendre la sélection comme ensemble de départ (ou comme cible pour les requêtes géométriques), ce mode permet d'enchaîner plusieurs requêtes de manière très efficace.
3. Créer une nouvelle couche : le résultat est disposé dans une nouvelle couche d'information, ou dans plusieurs nouvelles couches d'information si la requête prend plusieurs couches en entrée. Le nom de la nouvelle couche est alors formé du nom de la couche initial et du nom de la valeur définie dans la requête (dernier combobox), séparés par un espace souligné (underscore).

2.4. Le compositeur de requête

Le composeur de requête permet de définir cinq champs :

2.4.1. Champ Layer

Permet de sélectionner l'ensemble de départ sur lequel la requête sera effectuée. Cet ensemble peut être :

- Toutes les couches : la requête va balayer tous les objets de toutes les couches. Les couches pour lesquelles la requête n'a pas de sens sont ignorées.
- Sélection : les objets interrogés sont ceux qui sont sélectionnés dans la carte active. Ils peuvent appartenir à plusieurs couches. Ces objets peuvent aussi être le résultat d'une requête précédente, effectuée avec la case Sélectionnée cochée.
- Couches sélectionnées : interroge uniquement les couches sélectionnées dans le gestionnaire de couches de JUMP.
- COUCHE_UTILISATEUR : après ces trois sources "spéciales", vient l'énumération de toutes les couches utilisateur disponibles et pouvant être interrogées individuellement.

Avertissement

Chaque fois qu'une nouvelle couche est sélectionnée, la liste des attributs disponibles est remise à jour ainsi que les fonctions et les opérateurs disponibles. De la même façon, quand un attribut est changé, ou une fonction, les listes suivantes sont mises à jour et les valeurs présentées peuvent changer. Pour cette raison, il est recommandé de toujours créer ses requêtes en sélectionnant les champs de la gauche vers la droite.

2.4.2. Champ attribut

Dans cette liste déroulante figurent tous les attributs disponibles. Cette liste dépend du ou des layers choisis, ainsi que des options cochées dans le panneau d'options "filtrer les attributs".

Les noms d'attributs sont suivis de leur type géométrique entre parenthèses. Ils sont rangés par ordre alphabétique avec toutefois en tête de liste l'attribut commun (GEOMETRY).

La sélection d'un attribut détermine le choix des fonctions disponibles (liste déroulante suivante). Dans le cas des attributs de type caractère, une liste de valeurs possibles est dressée dans la liste déroulante **valeur**. Cette liste peut être exhaustive (cas des types d'attributs énumérés) ou indicative (dans le cas de textes quelconques, le programme va chercher les 12 premières valeurs possibles).

2.4.3. Champ fonction

L'attribut choisi pour effectuer la requête peut passer par une fonction avant que la condition finale ne soit évaluée. La liste des fonctions varie suivant le type de l'attribut sélectionné.

2.4.3.1. Fonctions géométriques

Les fonctions à votre disposition si vous utilisez la géométrie de l'objet pour effectuer le test sont :

- : laisser en blanc pour utiliser la géométrie telle que

- **longueur** : renvoie la longueur de la géométrie. C'est sur cette longueur que va porter le test
- **surface** : renvoie la surface de la géométrie. C'est sur cette surface que va porter le test
- **nombre de points** : renvoie le nombre de points total de la géométrie
- **nombre de parties** : renvoie 1 ou le nombre de parties dans le cas de géométries composites
- **buffer** : renvoie une géométrie qui est en fait une zone tampon autour de la géométrie de départ. Cette fonction a la particularité de prendre un argument. Pour pouvoir changer l'argument, la zone a été rendue éditable. Attention de bien respecter la syntaxe donnée en exemple
- **centroïde** : renvoie un point situé à l'intérieur de la géométrie (ou dessus dans le cas d'une géométrie linéaire)
- **vide** : renvoie "true" si la géométrie est vide, "false" dans le cas contraire
- **simple** : renvoie "true" si la géométrie est simple, "false" dans le cas contraire
- **valide** : renvoie "true" si la géométrie est valide, "false" dans le cas contraire

Les trois dernières fonctions font référence au "Simple Feature Specification" de l'OGC et à la documentation de l'API de JTS.

Les fonctions géométriques sont appliquées à l'attribut (GEOMETRY) et peuvent renvoyer une autre géométrie, une valeur numérique ou encore un booléen (true/false ou vrai/faux)

2.4.3.2. Fonctions numériques

Aucune fonction numérique définie.

2.4.3.3. Fonctions texte

- : le texte est laissé tel quel pour la comparaison
- **trim** : le texte est "nettoyé" de ses blancs initiaux et finaux. Cette fonction est très utile pour comparer des attributs provenant de fichiers shapefile et souvent complétés par des blancs quand il ne font pas la longueur maximum autorisée par le champ.
- **sous-chaîne** : permet d'extraire une chaîne de caractères à partir de la valeur d'attribut. Cette fonction peut prendre deux arguments, et pour cela, elle a été rendue éditable. Attention de bien respecter la syntaxe donnée en exemple dans le constructeur de requêtes. L'indice 0 correspond au premier caractère. Si la valeur d'attribut est "Seine-Saint-Denis", la fonction sous-chaîne(0,5) renvoie la chaîne "Seine"
- **longueur** : cette fonction renvoie la longueur de la chaîne de caractères.

Les fonctions texte sont appliquées à des attributs de type texte, caractère ou énuméré et peuvent renvoyer un autre texte ou une valeur numérique.

Astuce

Vous trouverez dans les opérateurs une manière bien plus puissante d'effectuer des recherches de chaînes de caractères. Il s'agit des opérateurs **match** et

contient qui utilisent les expressions régulières. L'expression régulière est alors définie dans la zone de texte **valeur** (voir § 2.4.4.3 et § 2.4.5.4)

2.4.4. Champ opérateur

La liste des opérateurs disponibles est déterminée par le type de l'attribut (géométrie/numérique/caractère) ou par le type renvoyé par la fonction utilisée si ce type est différent du type de l'attribut. Ainsi, les opérateurs peuvent se répartir de la façon suivante :

2.4.4.1. Opérateurs géométriques

- **intersecte** : sélectionne tous les objets intersectant l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).
- **contient** : sélectionne tous les objets contenant l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).
- **est inclu** : sélectionne tous les objets inclus dans l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).
- **à moins de** : il s'agit d'un opérateur paramétrable (la zone est éditable pour cette valeur particulière dont vous pouvez changer l'argument en prenant soin de respecter la syntaxe donner en exemple). Sélectionne tous les objets situés à moins de [argument] de l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).
- **touche** : sélectionne tous les objets touchant l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).
- **traverse** : sélectionne tous les objets traversant l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).
- **recouvre** : sélectionne tous les objets ayant un recouvrement avec l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).
- **est disjoint de** : sélectionne tous les objets disjoint de l'un ou l'autre des objets définis par la liste déroulante **valeur** (à savoir, la sélection, toutes les couches ou une couche au choix).

Avertissement

l'opérateur disjoint est trompeur, dans la mesure où il suffit à l'objet d'être disjoint d'un seul objet de l'ensemble cible pour être sélectionné. Il peut par conséquent touché ou intersecté d'autres objets de la cible !

2.4.4.2. Opérateurs numériques

Les opérateurs numériques sont :

- **=** : égal

- # : différent
- < : inférieur
- > : supérieur
- <= : inférieur ou égal
- >= : supérieur ou égal

Les entiers, les doubles et les décimaux sont tous traités de la même façon (convertis en double avant la comparaison)

2.4.4.3. Opérateurs de comparaison de chaînes

Les opérateurs permettant d'effectuer des comparaisons de chaînes sont :

- **égal** : test la stricte égalité entre l'attribut (ou l'attribut transformé par la fonction), et la chaîne entrée dans le champ **valeur** (voir ci-dessous). Si la case à cocher Sensible à la casse (voir 2.2) est cochée, la comparaison tiendra compte des majuscules.
- **différent** : c'est la fonction opposée à la précédente
- **commence par** : teste si les premières lettres de l'attribut ou de la chaîne résultante coïncident avec la chaîne entrée dans le champ **valeur**.
- **fini par** : test si les dernières lettres de l'attribut ou de la chaîne résultante coïncide avec la chaîne entrée dans le champ **valeur**. Il peut être prudent d'utiliser la fonction trim devant cet opérateur.
- **match** : il s'agit d'un opérateur très riche qui permet de comparer l'attribut à une expression régulière. Si l'attribut ou l'attribut transformé par la fonction coïncide dans sa totalité, à l'expression régulière, la condition est vérifiée. L'opérateur prend en compte l'option cochée dans la case Sensible à la casse. Vous trouverez quelques exemples d'expressions régulières ci-dessous (§ 2.4.5)
- **contient** : cet opérateur utilise comme le précédent les expressions régulières. A la différence du précédent, la condition est considérée comme vérifiée si une partie quelconque de la chaîne en entrée coïncide avec l'expression définie dans le champ **valeur**.
- **avant** : trouve tous les objets dont l'attribut sélectionné est situé avant la chaîne spécifiée dans **valeur**, suivant l'ordre alphabétique.
- **après** : trouve tous les objets dont l'attribut sélectionné est situé après la chaîne spécifiée dans **valeur**, suivant l'ordre alphabétique.

2.4.5. Valeur de référence

La valeur de référence peut être de différente nature en fonction de la nature de l'opérateur :

2.4.5.1. Références pour les opérateurs géométriques

Dans le cas d'opérateurs géométriques, la référence est constitué par l'ensemble des géométries présentes dans l'un des choix suivants :

- la sélection

- un autre objet de la couche
- un autre objet de la tâche (ou un objet quelconque d'une couche quelconque)
- un objet d'une couche précise

2.4.5.2. Référence pour les opérateurs numériques

Si l'opérateur prend des nombres en arguments, la valeur entrée dans le champ valeur doit être un nombre. Utiliser le "." comme séparateur décimal.

2.4.5.3. Référence pour les opérateurs booléen

Dans certains cas, les fonctions géométriques renvoient une valeur booléenne. Dans ce cas préci, les seuls opérateurs disponibles sont "égal" ou "différent" et les seuls valeurs possibles sont true (vrai) ou false (faux).

2.4.5.4. Référence pour les chaînes de caractères

Pour les chaînes de caractères, la référence est soit une chaîne (égal, différent, commence par, fini par, avant, après), soit une expression régulière (match, contient). Ni l'une ni l'autre n'ont besoin d'être mise entre guillemets.

Pour éviter la confusion, lorsque vous choisissez l'un de ces deux derniers opérateurs, le terme expression régulière apparaît dans la case **valeur**.

Voici quelques exemples qui montrent comment utiliser une expression régulière :

- match `\A\s+\S+\s+\Z` : chaîne contenant autre chose que des espaces (`\S+`) mais commençant (`\A`) ou finissant (`\Z`) par un ou plusieurs (+) espaces (`\s`)
- match `S(ain)?te[-\s]?Th[eée]r[eée]se` : trouve le nom Sainte-Thérèse écrit Ste Thérèse ou Sainte Thérèse, avec ou sans trait d'union et avec différentes formes d'accentuation.
- find `[\d]+([\.,][\d]+)?` : trouve les chaînes incluant un nombre entier ou décimal, utilisant le séparateur décimal point ou virgule

2.5. Lancer la requête

Une fois la requête entièrement composée, vous pouvez cliquer sur Valider, ce qui lancera son exécution.

Pendant l'exécution de la requête vous pourrez voir :

- Le nom de la couche en cours de traitement devant une patience indiquant la proportion d'objets passés en revu.
- Un message situé juste sous le composeur de requête, et indiquant que la requête est encours de traitement

A la fin du traitement, la patience revient à 0 et le message indique le nombre d'objets trouvés.

Parmi les boutons situés tout en bas de l'interface, vous trouverez un bouton Rafraichissement que vous devrez activer si votre modèle de données a évolué depuis la dernière requête (ajout ou suppression de tables).

A. Historique des changements

- **Version 0.1.0** (04 dec 2004)
 - Version initiale