



**Forum: WD 10**

**Topic: Recherche ( très souple )**

**Subject: Re: Recherche ( très souple )**

Posté par: drcharly93

Contribution le : 2/11/2007 10:44:17

Bonjour,

Citation :

ne différencie pas les caractères majuscules des caractères minuscules, ne tient pas compte des espaces situés avant et après la chaîne à tester, ne tient pas compte des caractères accentués minuscules, ne tient pas compte des espaces et des ponctuations à l'intérieur des chaînes.

C'est le fonctionnement par défaut de WinDev.

Pour vous en assurer aller dans l'éditeur d'analyse, clic droit sur le fichier en question, puis sélectionner l'option 'Description des rubriques'

vérifier enfin le bloc 'Paramètres de l'index et de la recherche pour les clés texte'(vous comprendrez qu'il vous faudra même pour une question de rapidité de la recherche définir cette rubrique comme 'clé avec doublon')

Vous trouverez les options référents à votre demande.

Vous Pouvez même utilisez la phonétique

**1) Récupération des mots**

**a) Procédure pour splittter une chaîne**

PROCEDURE Split( ChaîneSource est une chaîne, CaractereSeparateur est une chaîne, bSupprimeEspace est un booléen = Faux ) tbl est un tableau dynamique de 0 chaînes sListe est une chaîne = ChaîneSource sUnMot est une chaîne POUR TOUTE CHAÎNE sUnMot DE SListe SEPARÉE PAR CaractereSeparateur Si bSupprimeEspace = Vrai Alors sUnMot = SansEspace( sUnMot ) Fin TableauAjoute( tbl, sUnMot ) FIN RENVOYER tbl

**b) Méthode pour prendre différent séparateur**

//On remplace les points virgules par des virgules ValDifficultes = Remplace( ValDifficultes, ";", "," )  
//On remplace les deux points par des virgules ValDifficultes = Remplace( ValDifficultes, ":", "," ) //On remplace les espaces par des virgules ValDifficultes = Remplace( ValDifficultes, " ", "," ) //On peut continuer comme ça

**c) On alimente le tableau par un splittage de la chaîne transformée**

tblVar = Split( ValDifficultes, ",", Vrai )

Maintenant le tableau tblVar de mon précédent exemple est alimenté.

**2) Utilisation de la phonétique**

Pour que cela fonctionne il faut modifier la requête de départ exemple:

Citation :

```
SELECT
CLIENT.NumClient AS NumClient,
CLIENT.Civilité AS Civilité,
```

```
CLIENT.NomClient AS NomClient,  
CLIENT.DateNaissance AS DateNaissance  
FROM  
Client
```

par

```
sClientRecherche est une chaine = "dupon" Sql est une chaine Rs est une source de donnees Sql =  
[ SELECT CLIENT.NumClient AS NumClient, CLIENT.Civilité AS Civilité,  
WL.Phonétique(CLIENT.NomClient) As RubTest CLIENT.NomClient AS NomClient,  
CLIENT.DateNaissance AS DateNaissance FROM Client WHERE RubTest LIKE ] Sql +=  
"'" + WL.Phonétique(sClientRecherche) + "%'"
```

Modification de mon précédent l'exemple

Citation :

//On construit la condition

Pour Ind = 1 a Dimension( tblVar)

```
sCondition += RC + "[Difficultés] LIKE '%" + tblVar[Ind] + "%' "
```

Fin

```
//On construit la condition Pour Ind = 1 a Dimension( tblVar) sCondition += RC + "RubTest") LIKE  
&#039;" + WL.Phonétique(tblVar[Ind]) + "%'" Fin
```

Ce code reste toutefois théorique et n'a pas été testé