



Forum: WD 12

Topic: réplication base HF

Subject: Re: réplication base HF

Posté par: GoZeub

Contribution le : 18/12/2008 10:20:34

Une méthode pour sauvegarder un fichier complet (sans filtre) :

```
PROCEDURE Recup_Fichier(NomFichier) // Déclaration et initialisation des variables
NomCheminFichier est une chaîne = NomFichier+".txt" IdFichier est un entier ResEcrit est un
booléen = Vrai ResFermeFichier,nNbEnr est un entier sLesRubs,sMaRub,sMaLig sont des chaînes
sLesRubs=HListeRubrique(NomFichier) // Ouverture du fichier
fRepCrée(ComplèteRep(ComplèteRep(fRepExe())+"Sauvegarde "+DateSys())) IdFichier =
fOuvre(ComplèteRep(ComplèteRep(fRepExe())+"Sauvegarde "+DateSys())+NomCheminFichier,
foCréation+foEcriture) // Affichage du message d'erreur si l'ouverture n'a pas été
effectuée SI IdFichier = -1 ALORS Erreur(ErreurInfo(errMessage)) RETOUR FIN // Lecture du
premier enregistrement HLitPremier(NomFichier) // Autres enregistrements à lire ? Erreur
d'écriture ? TANTQUE HEnDehors = Faux ET ResEcrit = Vrai nNbEnr++
LIB_Detail=nNbEnr+" enregistrements traités" // Écriture des enregistrements ligne par ligne dans
le fichier texte sMaLig="" sMaRub = ExtraitChaîne(sLesRubs, rangPremier, RC) TANTQUE
sMaRub <> EOT SI sMaLig<>"" ALORS sMaLig+=TAB
sMaLig+="{NomFichier+"."+"sMaRub,indRubrique} sMaRub = ExtraitChaîne(sLesRubs,
rangSuivant, RC) FIN ResEcrit = fEcritLigne(IdFichier, sMaLig) // Lecture des enregistrements
suivants HLitSuivant(NomFichier) FIN // Affichage du message d'erreur si l'écriture
n'a pas été effectuée SI PAS ResEcrit ALORS Erreur(ErreurInfo(errMessage)) // Fermeture du
fichier ResFermeFichier=fFerme(IdFichier) SI ResFermeFichier = -1 ALORS // Affichage du
message d'erreur si la fermeture n'a pas été effectuée Erreur(ErreurInfo(errMessage))
FIN
```

Une méthode pour intégrer un fichier txt dans une Bdd (des modifications sont à prévoir pour adaptation) :

```
PROCEDURE Restaure_Fichier(NomFichier,sMonRep) // Déclaration et initialisation des variables
NomCheminFichier est une chaîne = NomFichier+".txt" IdFichier est un entier ResEcrit est un
booléen = Vrai ResFermeFichier,nNbEnr,nNumRub est un entier
sLesRubs,sMaRub,LigneLue,sValRub sont des chaînes tabRub,tabValeur sont des tableaux de 0
chaînes sLesRubs=HListeRubrique(NomFichier,hLstDétail) dh est une
chaîne=GetDateHeure(DateSys()+HeureSys()) sdReqDel est une Source de Données SI PAS
HExécuteRequêteSQL(sdReqDel,nom_connection,ModeDataUpdate,"DELETE FROM
"+NomFichier) ALORS Erreur(ErreurInfo()) RETOUR FIN HAnnuleDéclaration(sdReqDel) //
Ouverture du fichier IdFichier = fOuvre(ComplèteRep(sMonRep)+NomCheminFichier, foLecture) //
Affichage du message d'erreur si l'ouverture n'a pas été effectuée SI IdFichier =
-1 ALORS Erreur(ErreurInfo(errMessage)) RETOUR FIN LigneLue = fLitLigne(IdFichier) // Autres
lignes à lire ? Erreur de lecture ? TANTQUE LigneLue <> EOT ET LigneLue <> "" nNbEnr++
LIB_Detail=nNbEnr+" enregistrements traités" // Lecture des rubriques une par une nNumRub=0
tabRub=allouer tableau de 0 chaînes tabValeur=allouer tableau de 0 chaînes sMaRub =
```

```

ExtraitChaîne(sLesRubs, rangPremier, RC) TANTQUE sMaRub <> EOT  nNumRub++
  sValRub=ExtraitChaîne(LigneLue,nNumRub)  SI sValRub<>"" ALORS  SI
ExtraitChaîne(sMaRub,2)="T" ALORS  SI
Position(ExtraitChaîne(sMaRub,1),"date",0,SansCasse)>0 OU
Position(ExtraitChaîne(sMaRub,1),"duree",0,SansCasse)>0 OU
Position(ExtraitChaîne(sMaRub,1),"heure",0,SansCasse)>0 ALORS  SI
Majuscule(sValRub)="DATEMODIF" ALORS  sValRub=dh  SINON  SI Val(sValRub)=0
ALORS  sMaRub = ExtraitChaîne(sLesRubs, rangSuivant, RC)  CONTINUER  FIN
  sValRub=GetDateHeure(sValRub)  FIN  SINON  sValRub="&#039;"&sValRub+"&#039;"
  FIN  FIN  TableauAjoute(tabRub,ExtraitChaîne(sMaRub,1))
  TableauAjoute(tabValeur,sValRub)  FIN  sMaRub = ExtraitChaîne(sLesRubs, rangSuivant, RC)
FIN  SI Dimension(tabRub)=Dimension(tabValeur) ALORS
  Insert_Fichier(NomFichier,tabRub,tabValeur)  FIN  // Lecture des lignes suivantes du fichier
LigneLue = fLitLigne(IdFichier)  FIN  // Affichage du message d&#039;erreur si l&#039;écriture
n&#039;a pas été effectuée SI PAS ResEcrit ALORS Erreur(ErreurInfo(errMessage)) // Fermeture du
fichier ResFermeFichier=fFerme(IdFichier) SI ResFermeFichier = -1 ALORS // Affichage du
message d&#039;erreur si la fermeture n&#039;a pas été effectuée  Erreur(ErreurInfo(errMessage))
FIN

```

Une procédure appelé par la procédure de restauration :

```

PROCEDURE Insert_Fichier(NomFichier,tabRubriques,tabValeurs) sdReqInsert est une Source de
Données SQLCh est une chaîne nNbDim est un entier nNumDim est un entier  SQLCh="INSERT
INTO "+NomFichier+" ("  nNbDim=Dimension(tabRubriques) SI PAS nNbDim>0 ALORS RETOUR
POUR nNumDim=1 A nNbDim  SQLCh+=tabRubriques[nNumDim]  SI nNumDim<nNbDim ALORS
SQLCh+=","  FIN  SQLCh+=")  VALUES ("  nNbDim=Dimension(tabValeurs) SI PAS nNbDim>0
ALORS RETOUR POUR nNumDim=1 A nNbDim  SQLCh+=tabValeurs[nNumDim]  SI
nNumDim<nNbDim ALORS SQLCh+=","  FIN  SQLCh+=")"  SI PAS
HExécuteRequêteSQL(sdReqInsert,nom_connection,ModeDataUpdate,SQLCh) ALORS
Erreur(HErreurInfo())  RETOUR  FIN HAnnuleDéclaration(sdReqInsert)

```

Je peux pas faire plus désolé et bon courage