DepotFox

DepotFox est un utilitaire d'archivage et d'historique sécurisé pour tous les fichiers, intégrant la gestion des tuples : fichiers de développement fox, que ce soit les binaires (tables, bases de données, projets, formulaires, classes, rapports, étiquettes, menus) ou les fichiers plats (programmes, programmes de menu, requêtes), mais aussi tout type de fichier associé à son fichier de signature.

DepotFox crée un répertoire réseau virtuel et vous copiez/collez vos dossiers de developpement dans ce répertoire (15 niveaux de sous-dossiers autorisés).
DepotFox transfère les fichiers depuis ce répertoire vers une table SQL Server avec les informations d'arborescence, tout en associant de façon persistante les fichiers constituant un tuple. Cette association porte sur les binaires fox, et sur les signatures de fichiers.
À chaque transfert vers les tables d'archivage, DepotFox détecte les éventuelles modifications faites sur un élément et dans ce cas archive automatiquement la version précédente.

DepotFox est constitué d'un module serveur et d'un module client. Le module serveur utilise SQL Server 2016 (toutes éditions possibles, y compris Express). Le module serveur est créé par un programme VFP9. Le module client consiste en un unique objet de type Custom contenu dans un vcx.

Installation du module serveur

Prérequis

Vous devez impérativement disposer d'une version SQL Server 2016 (ou supérieure), et avoir activé FileStream en accès complet sur cette instance SQL. Vous trouverez les informations complètes sur cette activation ici : https://docs.microsoft.com/fr-fr/sql/relational-databases/blob/enable-and-configure-filestream

Installation

C'est le programme `InstallDepotFox.prg` qui installe le module serveur. Ce prg attend 3 paramètres, qui sont tous obligatoires :

- La chaine de connexion à l'instance SQL sur laquelle vous voulez installer la fonctionnalité DepotFox. Cette chaine doit être de la forme :
  `"Driver={SQL Server Native Client 11.0};`
  `Server=Machine\nomInstance;Database=master;Trusted_Connection=yes;"`
- Le nom de la base de données que vous voulez créer dans cette instance SQL, et qui hébergera la fonctionnalité
- Le nom du dossier FileStream pour l'accès Windows (c'est dans ce dossier virtuel que vous trouverez le sous-dossier DepotFox dans lequel vous copierez les fichiers fox à archiver pour historique

Le chemin complet du dossier virtuel partagé créé sur le serveur est mis dans une variable publique nommée `gcVirtualPath`

Utilisation du module Serveur

Le dossier virtuel partagé sur le réseau est immédiatement utilisable après l'installation du module serveur ; vous devez configurez les droits d'accès comme pour tout autre dossier Windows, le programme d'installation ne le fait pas par mesure de sécurité.

Pour l'utiliser, il suffit d'y coller les dossiers contenant les fichiers devant être archivés.

Lancement du module client

Pour utiliser un objet DepotFox instancié depuis la classe `DepotFox` qui se trouve dans foxhisto.vcx, vous disposez de trois approches à votre convenance (que vous utilisiez `CreateObject` ou `NewObject`).

- Méthode n°1

  Vous pouvez ouvrir la classe dans le générateur de classe de VFP, et écrire « en dur » les valeurs des propriétés indispensables :

  - `ChaineConnexion` (chaine de connexion complète)
  - `NomDatabase` (la base de données créée par l'installation)
  - `NomInstance` (le nom complet de l'instance SQL hébergeant la base de données)
  - `CheminDossier` (l'URI complète du dossier de dépôt)
  - `IsConnectStringValid` à .T.

  Vous instanciez votre objet sans aucun paramètre, et il est immédiatement utilisable.

- Méthode n°2

  Vous instanciez votre objet depuis la classe en lui passant le premier paramètre, la chaine de connexion complète contenant le nom de la database sur le serveur SQL. La méthode `Init` se chargera de vérifier cette chaine, et de récupérer les informations requises sur le serveur.

- Méthode n°3

  Vous instanciez votre objet depuis la classe en lui passant les deux derniers paramètres, c'est à dire le nom de la database et le nom complet de l'instance SQL. La méthode `Init` se chargera de construire la chaine de connexion nécessaire, et de récupérer les informations requises sur le serveur.

Utilisation du module client

DepotFox gère les fichiers fox et les fichiers non fox. Pour chaque traitement, vous disposez de méthodes pour les fichiers fox, de méthodes pour les fichiers non fox, et de méthodes pour tous les fichiers. Vous choisissez celle qui convient le mieux à votre travail.

- **Types de fichiers non fox**

Vous devez définir quels fichiers non fox seront pris en compte par DepotFox dans le répertoire virtuel de dépôt. Un fichier est défini par son extension ; les différentes extensions sont regroupées en types, et peut être activée ou désactivée. La liste des extensions définies est visible dans la propriété `TypesFichiersNonFox`.

  - Pour insérer/modifier/supprimer une seule extension, utilisez les méthodes `InsertTypeFichierNonFox`, `UpdateTypeFichierNonFox`, `DeleteTypeFichierNonFox`.
  - Pour insérer/modifier/supprimer un ensemble d'extensions, effectuez les opérations voulues dans le curseur TYPES_NON_FOX obtenu par la méthode `SetTypesFichiersNonFox`, puis mettez à jour les données avec la méthode `UpdateTousTypesFichiersNonFox`

- **Archivage et historisation**

Utilisez les méthodes `ArchiveVersHistoriqueFox`, `ArchiveVersHistoriqueFichiers`, ou `ArchiveVersHistoriqueTous`. C'est aussi simple que ça ! Les fichiers nouvellement déposés dans le

répertoire virtuel sont ajoutés à la table de stockage appropriée, ceux qui ont été modifiés mettent à jour la table d'historique.

- **Obtention de l'historique des données**

Utilisez les méthodes `GetHistoriqueFox (DateHeureDebut, DateHeureFin)` et `GetHistoriqueNonFox (DateHeureDebut, DateHeureFin)`. Le résultat est retourné dans un curseur VFP, dont le nom commence toujours par HistoriqueFox_ ou HistoriqueFichiers_

  - Historique des fichiers Fox
    - les champs `Fox_Nom_Table`, `Fox_TypeTable`, `Fox_Chemin` permettent d'identifier le binaire Fox
    - les champs `Fox_Data`, `Fox_Index`, `Fox_Memo` contiennent les fichiers fox en question (voir plus bas pour leur extraction)
    - les champs `Debut` et `Fin` donnent les dates limites de la version contenue dans l'enregistrement
  - Historique des fichiers Non Fox
    - les champs `Fichier_Nom`, `Fichier_Type`, `Fichier_Chemin` permettent d'identifier le fichier, le champ `Signature_Nom` identifie la signature éventuelle.
    - le champ `Fichier_Data` contient le fichier en question, `Signature_Data` le contenu de la signature si elle existe (voir plus bas pour leur extraction)
    - les champs `Debut` et `Fin` donnent les dates limites de la version contenue dans l'enregistrement
  - Le contenu dépend du nombre de paramètres (type `Date` ou `DateTime`) passés :
    - Sans aucun paramètre, vous obtenez tout l'historique pour tous vos fichiers.
    - Avec les deux paramètres de valeurs différentes, vous obtenez l'historique contenu dans cet intervalle bornes incluses
    - Avec un seul des deux paramètres, vous obtenez l'historique sur l'intervalle demandé
      - date de fin spécifiée, c'est « voir mon travail jusqu'à telle date »
      - date de début spécifiée, c'est « voir mon travail depuis telle date »
    - avec la même valeur pour les deux paramètres, vous obtenez l'historique sur cette date exclusivement.

- **Extraction depuis l'historique**

Positionnez-vous sur l'enregistrement du curseur d'historique contenant la version à restaurer sur votre disque, et appelez la méthode `ExtraitFichier (destination)`. Les fichiers sont recréés à la destination passée, dans la même arborescence qu'au stockage. Si aucune destination n'est indiquée, un dossier sera créé dans le dossier VFP courant.

- **Sécurité**

Vous pouvez déconnecter le répertoire virtuel de dépôt à tout moment en utilisant la méthode `SetDossierVirtuel (Actif_ou_Inactif)`.
Cette déconnexion est immédiate, et persistante même en cas d'arrêt du serveur. Les données contenues dans ce répertoire virtuel sont conservées, et sont à nouveau disponibles dès que vous rétablissez la connexion en appelant la méthode avec le paramètre "Actif".

- **Options**

Un objet Options est automatiquement instancié en tant que propriété de l'objet créé depuis DepotFox, mais vous pouvez également l'instancier séparément (les 3 mêmes méthodes que l'objet DepotFox).

- ***Version***
  Vous pouvez attribuer des numéros de version personnalisée à vos fichiers stockés, plusieurs versions physiques d'un même fichier pouvant porter le même numéro de version personnalisée. Une version se caractérise par son numéro (chaine de caractère de la forme 1.1.2.n…) et par son horodatage de début (les versions physiques postérieures ou égales à cet horodatage porteront ce numéro logique)
  - Visualisation des numéros de version
    utilisez une des méthodes `GetFichiersFoxVersions`, `GetFichiersNonFoxVersions`, ou `GetTousFichiersVersions`.
    Ces méthodes vous retournent les fichiers avec leur éventuel numéro de version personnalisée.
  - Insertion – modification – suppression de numéros de version
    pour un même fichier (même nom, même type, même dossier), deux numéros de versions différents ne peuvent pas avoir le même horodatage de début, et un numéro de version ne peut pas avoir deux horodatages de début.
    - Pour insérer/modifier/supprimer un seul numéro de version, utilisez les méthodes `InsertVersionFoxTable` et `InsertVersionFichiers`, `UpdateVersionFoxTable` et `UpdateVersionFichiers`, ou `DeleteVersionFoxTable` et `DeleteVersionFichiers`.
    - Pour insérer/modifier/supprimer un ensemble de numéros de version, effectuez les opérations voulues dans les curseurs VersionsFox et VersionsNonFox obtenus par les méthodes `SetVersionsFox` et `SetVersionsNonFox`, puis mettez à jour les données avec les méthodes `UpdateTousFichiersFoxVersions` et `UpdateTousFichiersNonFoxVersions`
- ***Projet***
  Les projets sont des catégories abstraites organisées en arborescences, auxquelles vous pouvez rattacher vos binaires. Un fichier stocké en table d'historique peut être rattaché à autant de projets et/ou sous-projets que vous le souhaitez.

  Gestion des Projets

  - Visualisation des projets
    utilisez la méthode `GetProjets` pour obtenir la liste des projets dans un curseur VFP. Ce curseur vous présente le niveau dans l'arborescence, le nom du projet, et un commentaire optionnel.
  - Insertion – modification – suppression de projets
    Les champs `Niveau` et `Nom_Projet` sont obligatoires. Un Niveau est une chaine de caractères de type /1/1/2/…
    Dans le curseur obtenu par la méthode GetProjets, effectuez les opérations voulues, puis mettez à jour les données avec la méthode `UpdateTousProjets`.

  Rattachements fichiers/projets

  - Recherche et Visualisation des rattachements des fichiers aux projets
    Utilisez la méthode `SearchTablesFichiersProjets` pour obtenir une vue des rattachements. Cette méthode attend 3 paramètres optionnels :
    - 1er paramètre NOM_PROJET_EGALE
    - 2ème paramètre TYPE _IN (pour les fichiers fox, vous pouvez passer aussi bien les types longs que les extensions, regardez les propriétés `TypesFichiersFox` et `TypesFichiersFoxCourts`)
    - 3ème paramètre NOM_ LIKE
    - Si vous passez plusieurs paramètres, ils sont combinés par AND

        - Aucun paramètre, vous obtenez la liste de tous les rattachements
    - Insertion – modification – suppression de rattachements
        - Pour insérer/modifier/supprimer un seul rattachement, utilisez les méthodes `InsertProjetFoxTable` et `InsertProjetFichiers`, `UpdateProjetFoxTable` et `UpdateProjetFichiers`, ou `DeleteProjetFoxTable` et `DeleteProjetFichiers`.
        - Pour insérer/modifier/supprimer un ensemble de rattachements, effectuez les opérations voulues dans les curseurs ProjetsFox ou ProjetsNonFox obtenus par les méthodes `SetProjetsFox` et `SetProjetsNonFox`, puis mettez à jour les données avec les méthodes `UpdateTousFichiersFoxProjets` et `UpdateTousFichiersNonFoxProjets`

- **Maintenance du module serveur**
    - Effacement de l'historique

    Pour effacer définitivement tout l'historique sur le serveur, utilisez les méthodes `ZapHistoriqueFox` et `ZapHistoriqueFichiers`

    - Sauvegarde des databases

    Comme pour toute base de données SQL Server vous devez impérativement les sauvegarder, sinon les fichiers des transactions (les mdf de Log) vont croitre indéfiniment ! Utilisez simplement la méthode `SauvegardeSQL` pour effectuer cette opération.

    Les propriétés `TailleMDFdepot`, `TailleMDFarchive`, `TailleLOGdepot`, `TailleLOGarchive`, `DateDerniereSauvegardeSQLdepot` et `DateDerniereSauvegardeSQLarchive` vous donnent les informations dont vous avez besoin pour décider de la nécessité d'une sauvegarde.

    Les propriétés `NbFichiersDansDepot`, `NbFichierDansArchive`, `NbFichiersDansHistorique` vous seront parfois utiles.

    Ces propriétés sont mises à jour lors de l'Init, et vous pouvez forcer leur actualisation à tout moment avec la méthode `SetAllSQLprop`.

Questions-Réponses

- À quoi sert l'archivage des tables ? est-ce que je peux utiliser DepotFox pour archiver les données ?
  - Techniquement, DepotFox archive les dbfs avec leur cdx et leur memo s'ils existent. Vous pouvez donc archiver n'importe quel dbf. Pensez simplement à surveiller la taille de la database SQL si vous archivez vos dbf de données.

- Quelles sont les signatures traitées par DepotFox ?
  - DepotFox gère l'association d'un fichier avec sa signature externe en fichier xml conforme au format XADES défini par le W3C (http://www.w3.org/TR/XAdES/).

- Est-ce que je peux installer DepotFox sur un serveur distant sur mon LAN (pas la même machine que celle de développement) ?
  - Oui, bien sur

- J'ai installé le module serveur sur un serveur distant, et je n'arrive pas à instancier l'objet client, que se passe-t-il ?
  - Il est très probable que vous n'avez pas configuré les droits d'accès au répertoire virtuel ; le client DepotFox vérifie qu'il a bien accès à ce dossier dans l'init. En cas d'échec de l'init, un objet de type exception est créé en variable publique, vous verrez le détail de l'erreur.

- Est-ce que VFP doit être installé sur la machine qui héberge le serveur SQL de DepotFox ?
  - Non, le module serveur de Depotfox n'utilise pas VFP

- Est-ce que je suis obligé d'utiliser la même chaine de connexion pour l'install et pour l'utilisation de l'objet client ?
  - Non, ce n'est ni nécessaire ni même recommandé : pour la connexion client, il est préférable d'utiliser une connexion qui dispose uniquement des droits SELECT, INSERT, UPDATE, EXEC sur les bases de données du serveur

- Notre DBA interdit la création des bases de données depuis un code client ; comment faire ?
  - Transmettez au DBA les 2 scripts SQL fournis avec DepotFox, pour qu'il puisse procéder lui-même à cette installation et aux paramétrages de sécurité d'accès et de droits.

- Peut-on avoir plusieurs serveurs DepotFox ?
  - Oui, un objet client peut s'adresser à autant de serveurs que vous le voulez ; il suffit de modifier les valeurs des propriétés de connexion, et de lancer les méthodes que vous trouverez dans l'Init.

- Pourquoi y a-t-il deux bases de données pour DepotFox ?
  - Une de ces deux bases est dédiée à la table de répertoire virtuel, ceci pour renforcer la sécurité ; en cas de menace d'intrusion, il suffit de déconnecter le répertoire virtuel en passant le paramètre "Inactif" à la méthode SetDossierVirtuel(), et vous conservez l'usage de l'archive avec son historique.

- Nous ne voulons pas avoir de dossier partagé en permanence, pour des raisons de sécurité. Comment faire ?
  - La déconnexion du répertoire virtuel est persistante ; déconnectez le répertoire virtuel en passant le paramètre "Inactif" à la méthode SetDossierVirtuel(), reconnectez-le uniquement quand vous voulez approvisionner le dépôt, puis déconnectez à nouveau.

- J'ai supprimé un fichier dans le répertoire virtuel, est-ce qu'il sera supprimé de l'archive et/ou de l'historique quand je ferai un ArchiveVersHistorique ?
  - Non, une fois qu'un fichier est en archive historisé, il n'est pas supprimé

- Est-il possible de modifier le contenu des fichiers dans l'historique ?
  - Non, la table d'historique est en lecture seule, aucune UPDATE n'est autorisé par SQL Server sur ce type de table

- Est-il possible de supprimer la table d'archivage ou la table d'historique ?
  - Non, ces deux tables sont protégées par le moteur SQL, et ne peuvent pas être supprimées

- Que se passe-t-il quand je lance une commande ZapHistorique ? y a-t-il une trace de cette commande ?
  - Oui, toutes les demandes de modification de l'historique sont enregistrées automatiquement dans une table non modifiable et non supprimable

- Est-ce que je peux utiliser directement le répertoire virtuel comme répertoire de travail VFP, y modifier directement mes fichiers fox ?
  - Non, les verrous que VFP pose pour ouvrir et traiter les fichiers de développement sont contradictoires avec ceux que SQL va poser en accédant aux API qui lui donnent accès à ce répertoire virtuel. Vous devez uniquement recopier les fichiers dans le répertoire virtuel.

- Je viens de copier des fichiers modifiés et des nouveaux fichiers, et de demander un ArchiveVersHistorique. Comment savoir combien de fichiers ont été traités ?
  - Interrogez les valeurs des propriétés NbNouveauxFichiersDansHistorique et nbNouveauxFichiersDansDepot

- Les projets du module Options sont-ils identiques aux projets de VFP ?
  - Non, ils sont totalement différents. Vous créez et organisez les projets des options à votre convenance ; vous pouvez par exemple avoir des zones géographiques, des secteurs d'activité professionnelle, des organisations de ressources humaines, etc… Imaginez plutôt les projets comme des mots-clés dotés d'une arborescence.

Historique et versions

25/01/2018	3	Intégration de tous les types de fichiers avec gestion des signatures associées
08/11/2017	2.2	Intégration des fichiers plats (prg, mpr, qpr) correction du bug sur les noms de fichier contenant des .
09/09/2017	2.1	Implémentation des projets hiérarchisés et des numéros de version
13/08/2017	1.2	Séparation en deux database, ajout d'une table de log pour les zap de l'historique
05/08/2017	1.1	traitement des GRANT dans l'install serveur, gestion du contexte du CREATE, suppression de la vérification d'accès au dossier virtuel
02/08/2017	1.0	fonctionnelle initiale