

StructNamesPack 2.0

Avril 1999.

Ce plug-in pour Quatrième Dimension™ contient des routines permettant de charger dans un tableau (alpha ou texte) les noms des méthodes projet, des énumérations, ou des formulaires d'une table.

Il sera ainsi possible, par exemple, de récupérer la liste des méthodes projet pour reproduire le dialogue "Exécuter une méthode..." du mode Utilisation directe. Vous pourrez également exporter/importer toutes vos énumérations au sein d'une méthode générique (en utilisant une boucle sur leurs noms chargés par le plug-in). Jusqu'à maintenant, pour faire cela, il fallait maintenir en permanence la liste des méthodes projet et des énumérations, retirer les méthodes supprimées, ajouter les nouvelles, bref, la maintenance d'une telle liste était un peu fastidieuse.

Contexte d'utilisation:

- 4D à partir de la version 6
- Plate-formes: Mac et Windows
- Structure: Interprétée, compilée, intégrée à 4D Engine
- Pour 4D Server: les routines doivent être exécutées sur le serveur (reportez-vous à la base de test et à l'exemple plus loin)

SYNTAXE

\$erreur:=**sn_Get methods names** (tabNoms)

\$erreur:=**sn_Get list names** (tabNoms)

\$erreur:=**sn_Get forms names** (tabNoms;numeroDeTable)

tabNoms: Tableau alpha ou texte

numeroDeTable: Entier. Numéro de la table dont on veut récupérer les formulaires.

\$erreur: Entier long. Code d'erreur retourné par la routine (voir plus loin)

Ces trois routines remplissent donc le tableau tabNoms avec:

- Les noms de toutes les méthodes projet,
- Les noms de toutes les énumérations,
- Les noms de tous les formulaires de la table numeroDeTable. Notez qu'il n'est pas possible de récupérer les noms des formulaires de sous tables.

Si tabNoms est un tableau alpha, l'externe tronquera les noms si nécessaire, sans retourner d'erreur.

Codes d'erreur

0: Pas d'erreur

-30 000: Le tableau n'est ni Alpha ni Texte.

-30 001: Le numéro de table est invalide (pour GetFormNames)

-30 002: Impossible de charger la table des méthodes, des énumérations ou des formulaires

-30 003: à partir de 4D version 6 uniquement

-30 004: Erreur lors du chargement de la structure

-30 005: ID de la table incorrect (Erreur interne au package)

-30 006: package non enregistré, numéro de licence incorrect

-30 007: ne fonctionne pas sur 4D Client

Tout autre numéro correspond à une erreur retournée par le système (erreur mémoire, erreur du gestionnaire de fichiers, ...)

Avec 4D Server

Le plug-in ayant besoin d'accéder au fichier de structure, un 4D Client ne peut l'appeler. Pour récupérer les noms des objets en mode client-serveur, il faut utiliser une procédure stockée.

Voici un exemple, parmi de nombreuses autres possibilités, de procédure stockée qui récupère dans un tableau alpha 255 les noms des objets, que l'on soit en monoposte ou en Client/Serveur.

Le principe est le suivant: Si l'on est en monoposte, le plug-in est directement appelé pour remplir le tableau. Si on est sur 4D Client, on utilise une procédure stockée. Pour faire cet échange en client serveur, il faut utiliser quelques variables interprocess sur le serveur.

La méthode base "Sur démarrage serveur" contiendra donc les déclarations suivantes:

```
`-----  
` Méthode base "Sur démarrage serveur"  
`-----
```

```
C_ENTIER LONG(◇getNames_Error) ` l'erreur retournée  
TABLEAU APHA(255;◇getNames_TabNoms;0) ` Le tableau dans lequel charger les noms  
C_BOOLEEN(◇getNames_ServerDone; ◇getNames_ClientDone) ` drapeaux d'attente
```

La méthode "GetTheNames" est celle utilisée pour remplir le tableau (qui devra être de type Alpha 255 dans notre exemple). Nous utilisons un des paramètres comme "sélecteur" permettant de savoir si l'on veut les noms des méthodes, des énumérations, ou des formulaires.

Selon que l'on est sur 4D Client ou non, elle renvoie l'appel sur le serveur ou exécute directement le plug-in

```
`-----  
` Méthode projet "GetTheNames"  
`-----
```

```
` $err:=GetTheNames(selecteur;->leTableau{;numTable})
```

```
` erreur renvoyée
```

```
C_ENTIER LONG($0)
```

```
` sélecteur: 1 = méthodes, 2 = énumérations, 3 = formulaires
```

```
C_ENTIER LONG($1)
```

```
` tableau à remplir
```

```
C_POINTEUR($2)
```

```
` numéro de table pour les formulaires
```

```
C_ENTIER LONG($3)
```

```
C_ENTIER LONG($serverProcess)
```

```
C_BOOLEEN($done)
```

```
TABLEAU ALPHA(255;$zeNames;0)
```

```
` Si on est sur 4D Client...
```

```
Si (Type application=4D_Client)
```

```
` Poser un sémaphore: il ne faudrait pas que plusieurs clients
```

```
` accèdent au même moment à cette méthode, sinon l'un
```

```
` risquerait de mettre le drapeau ◇getNames_ServerDone
```

```
` à Vrai alors que l'appel pour un autre client n'est pas terminé
```

```
Tant que (Semaphore("IIEtaitUnPetitNavire"))
```

```
APPELER 4D
```

```
Fin tant que
```

```
` lancer la procédure stockée
```

```
Si (Nombre de parametres<3)
```

```
$serverProcess:=Executer sur serveur("GetThingsOnServer";(32*1024);"GetStructNames";$1)
```

```
Sinon
```

```
$serverProcess:=Executer sur
```

```
serveur("GetThingsOnServer";(32*1024);"GetStructNames";$1;$3)
```

```
Fin de si
```

```
Si ($serverProcess<0)
```

```
` Attente que le serveur ait récupéré les noms (cf "GetThingsOnServer")
```

\$done:=Faux

Repeter

ENDORMIR PROCESS(Numero du process courant;1)

LIRE VARIABLE PROCESS(\$serverProcess;◇getNames_ServerDone;\$done)

Jusque (\$done)

` Lire l'erreur

LIRE VARIABLE PROCESS(\$serverProcess;◇getNames_Error;\$err)

Si (\$err=0) | (\$err=-30006) ` -30 006 = pas de licence: mode version de démonstration.

` Lire le tableau des noms

LIRE VARIABLE PROCESS(\$serverProcess;◇getNames_TabNoms;\$2->)

Fin de si

` Terminer la procédure stockée

ECRIRE VARIABLE PROCESS(\$serverProcess;◇getNames_ClientDone;**Vrai**)

` renvoyer l'erreur

\$0:=\$err

Sinon

` Mauvais numéro de process sur le server...

\$0:=-1 ` ...retourner une erreur quand même...

Fin de si

` Place nette

EFFACER SEMAPHORE("IIEtaitUnPetitNavire")

Sinon

Au cas ou

: (\$1=1)

\$0:=*sn_Get methods names* (\$2->)

: (\$1=2)

\$0:=*sn_Get lists names* (\$2->)

: (\$1=3)

\$0:=*sn_Get forms names* (\$2->,\$3)

Fin de cas

Fin de si

La méthode "GetThingsOnServer" est donc exécutée sur le serveur. Elle récupère les noms voulus dans le tableau interprocess ◇getNames_TabNoms, et place les drapeaux de tests

` -----

` Méthode projet "GetThingsOnServer"

` -----

` GetThingsOnServer(selecteur;numTable)

C_ENTIER LONG(\$1) ` sélecteur. 1: méthodes, 2 énumérations, 3: formulaires

C_ENTIER LONG(\$2) ` numéro de table pour les formulaires

` initialisations

TABLEAU ALPHA(255;◇getNames_TabNoms;0)

◇getNames_Error:=0

◇getNames_ClientDone:=Faux

◇getNames_ServerDone:=Faux

` récup. des noms

Au cas ou

: (\$1=1)

◇getNames_Error:=*sn_Get methods names* (◇getNames_TabNoms)

: (\$1=2)

◇getNames_Error:=*sn_Get lists names* (◇getNames_TabNoms)

: (\$1=3)

Si (Nombre de parametres=2)

◇getNames_Error:=**sn_Get forms names** (◇getNames_TabNoms;\$2)

Sinon

` renvoyer une erreur: "mauvais appel à GetThinsOnServer"

◇getNames_Error:=-10 ` ou autre chose...

Fin de si

Sinon

` renvoyer une erreur: "mauvais sélecteur pour GetThinsOnServer"

◇getNames_Error:=-11 ` ou autre chose...

Fin de cas

` Informer le client qu'on a exécuté le plug-in

◇getNames_ServerDone:=**Vrai**

` Attendre que lui même nous ait pillé

Repeter

ENDORMIR PROCESS(Numero du process courant;1)

Jusque (◇getNames_ClientDone)

` place nette

TABLEAU ALPHA(255;◇tA255_ServerNames;0)

◇getNames_Error:=0

◇getNames_ClientDone:=Faux

◇getNames_ServerDone:=Faux

ENREGISTREMENT

StructNamesPack est édité par **OSMOSE Éditeur**.

OSMOSE Editeur

33, Av. Jean Monnet

13410 LAMBESC

Tel.: 04 42 92 83 55

Fax: 04 42 92 83 27

web: <http://osmose.net>

eMail: editeur@osmose.net

Auteur: Thibaud Arguillere. thibaud@osmose.net

Et distribué par

e-Node SA

rue cantelaudette

za de la gardette

33310 LORMONT

Tel : +33 (0) 556 315 735

Fax : +33 (0) 556 778 429

Web : <http://e-node.com/>

E-Mail : sales@e-node.com

Son coût est de 60 Euros par plate-forme, 90 Euros pour les deux.

La licence est "Hassle-Free", il en faut une par développeur.

Lorsque vous aurez effectué votre règlement, vous recevrez un numéro de licence par plate-forme, qu'il faudra passer en paramètre à la routine **sn_Register**:

\$erreur:=sn_Register("licenceMac";"licenceWindows")

Tant que le produit n'aura pas été enregistré, l'erreur N°-30 006 vous sera retournée, et les tableaux seront tronqués aux 10 premières méthodes trouvées et aux 5 premières énumérations/ premiers formats trouvés

Cette routine doit être placée avant toute utilisation des autres routines du package, sinon il fonctionnera en mode de démonstration. Le plus judicieux est bien entendu de la placer dans la méthode base "Sur ouverture".