

Guide d'intégration



DINKEYPRO

Protection de logiciels VISUAL LISP avec DinkeyPRO/FD

AutoCAD®
Visual LISP® – AutoLISP® - VLISP®

Contact Commercial :
Tél. : 02 47 35 70 35
Email : com@aplika.fr

Contact Technique :
Tél. : 02 47 35 53 36
Email : support@aplika.fr

Version 1.00 du 26/05/2011



aplika

La Foltière - 37270 AZAY SUR CHER
Tél. 33(0)2 47 35 70 35 - Fax 33(0)2 47 35 70 25 - e-mail : aplika@aplika.fr

Guide d'intégration DinkeyPRO/FD VISUAL LISP



1. Introduction	3
2. Enregistrer dpcom32.dll sous Windows.....	3
3. Appel de la protection	3
3.1. Intégration du code à votre application – Test de la protection	3
3.2. Fonctions Get de DPCOM32.DLL.....	5
3.3. Code exemple Visual Lisp.....	6
3.4. Tableau des valeurs de FLAGS.....	7
3.5. Protection du module DPCOM32.DLL	7
4. Déploiement de votre application	10

Guide d'intégration DinkeyPRO/FD VISUAL LISP

1. Introduction

Afin de vous aider dans les premiers pas de la pose de protection sur vos logiciels VISUAL LISP, vous trouverez dans ce guide de prise en main rapide l'essentiel des commandes et fonctionnalités pour la bonne utilisation des clés DinkeyPRO/FD.

Méthode utilisée : API avec appel de l'ActiveX `dpcom32.dll`.

Pour accéder aux dongle DinkeyPRO ou FD à partir d'AutoCAD, vous devrez utiliser une DLL ActiveX avec Visual LISP. Cette librairie est appelée `dpcom32.dll`. Elle est située dans le dossier 'Modules' du répertoire d'installation du kit développeur DinkeyPRO/FD.

Pour protéger votre application vous devrez donc protéger `dpcom32.dll` en utilisant `DinkeyADD`, en spécifiant la méthode API comme méthode de protection.

Pour utiliser le contrôle ActiveX `dpcom32.dll`, vous devez préalablement l'enregistrer sous Windows.

2. Enregistrer `dpcom32.dll` sous Windows

Pour enregistrer la DLL sous Windows, cliquez sur *Démarrer -> Exécuter*, puis saisissez `regsvr32 "<chemin de la DLL>\dpcom32.dll"`

Pour la retirer, cliquez sur *Démarrer -> Exécuter*, puis saisissez `regsvr32 "<chemin de la DLL>\dpcom32.dll" /U`

Sous OS Windows 64 bits, vous devrez utiliser `regsvr32`, situé dans le dossier `Windows\systemwow64`. Ceci, au lieu du `regsvr32`, par défaut, qui inscrirait alors la DLL 32 bits comme DLL 64 bits (à noter que `systemwow64` est dédié aux fonctionnalités 32 bits, même si son nom peut prêter à confusion).

Le contrôle Active X `dpcom32.dll` devra être installé et déclaré sur les postes des utilisateurs. Vous pourrez automatiser cette installation grâce à des programmes d'installation qui proposent normalement ce type de fonctionnalité. En cas de difficulté, notre service technique pourra vous aider sur ce point.

3. Appel de la protection

3.1. Intégration du code à votre application – Test de la protection

Le contrôle ActiveX contient en interne une déclaration de structure DRIS (structure de la zone de données sécurisée du dongle). Le principe consiste à réaliser un test de protection afin que ce DRIS soit initialisé avec les valeurs de la clé connectée. Vous pourrez alors faire des appels aux différentes fonctions Get décrites ci-après qui permettront de faire une lecture des différents paramètres de la clé.

Ces valeurs seront actualisées lors de chaque nouveau test de protection et resteront valides et accessibles entre chaque test.

La sécurité de votre code peut être renforcée par l'utilisation des fonctions `ProtCheckWithAlg`. N'hésitez pas, après avoir pris en main et testé les fonctions simples de test de la clé, à enrichir votre protection avec ces fonctionnalités.

Cette méthode de protection appliquées aux programmes AutoCAD est moins sûre que pour d'autres langages. En effet, les limitations du langage AutoLisp ne permettent pas l'utilisation, par exemple, du cryptage du DRIS ou des données de la clé. Il restera malgré tout très difficile de contourner cette protection.

Notre exemple est simplement un guide d'utilisation des fonctions de l'ActiveX. Nous vous invitons à renforcer la protection de votre application en ajoutant des tests à différents endroits dans votre code, et, éventuellement, à vous reporter à notre manuel, au chapitre 'Améliorez votre Protection'.

Vous trouverez ci-après la liste des APIs de dpcom32.dll avec un exemple d'appel en Visual Lisp.

(Pour plus d'informations sur la structure DRIS, reportez-vous au manuel, chapitre "Structure" section "Structure DRIS").

ProtCheck

Syntaxe :

```
(ProtCheck flags execs_decrement alt_prog_name)
```

Retour :

return code (0 = succès, sinon numéro d'erreur)

Remarques :

Réalise un test simple de la protection.

Consulter le tableau ci-après pour les correspondances des valeurs Flags. Vous devez toujours indiquer execs_decrement et alt_prog_name même s'il n'est pas utilisé un appel typique devrait être : "ProtCheck" 0 0 ""

ProtCheckWithAlg

Syntaxe :

```
(ProtCheckWithAlg flags alg_number var_a var_b var_c var_d var_e var_f var_g var_h )
```

Retour :

return code (0 = succès, sinon numéro d'erreur)

Remarques :

Réalise un test de la protection et exécute l'algorithme utilisateur repéré par 'alg_number' avec les variables 'var_a'...'var_h'.

Appelez (GetAlgAnswer) pour obtenir le résultat de cet algorithme après un appel réussi de cette fonction.

WriteString

Syntaxe :

(WriteString flags offset DataString)

Retour :

return code (0 = succès, sinon numéro d'erreur)

Remarques :

Réalise un test de la protection et écrit une chaîne dans la zone de données de la clé Dinkey à partir de l'octet indiqué dans l'entrée offset.

ReadString

Syntaxe :

(ReadString flags offset length)

Retour :

return code (0 = succès, sinon numéro d'erreur)

Remarques :

Réalise un test de la protection et lit une chaîne dans la zone de données de la clé Dinkey à partir de l'octet indiqué dans l'entrée de offset. Appelez (GetDataRead) pour obtenir le résultat de cet algorithme après un appel réussi de cette fonction.

3.2. Fonctions Get de DPCOM32.DLL

Après avoir appelé chacune des fonctions ProtCheck, ProtCheckWithAlg, WriteString ou ReadString, vous pourrez alors réaliser des appels aux fonctions Get décrites ci-après. Elles vous permettront d'obtenir les données stockées dans le DRIS de l'ActiveX.

Ces fonctions retourneront -2 (ou "-2") si un test de protection n'a pas préalablement été effectué. Elles retourneront -3 (ou "-3") si un test a été réalisé mais a échoué.

(Pour plus d'informations sur la structure DRIS reportez-vous au manuel chapitre "Structure" section "Structure DRIS").

Chacune de ces méthodes ne nécessite aucun paramètre en entrée.

GetExtendedError - Retourne le code d'erreur étendue (uniquement si l'appel à la protection a échoué)

GetType - Retourne le type de dongle détecté. Valeur numérique – Voir documentation DRIS.

GetModel - Retourne le modèle de dongle détecté. Valeur numérique – Voir documentation DRIS.

GetSDSN - Retourne le SDSN – Numéro de série développeur.

GetProdCode - Retourne le code produit dans une chaîne.

- GetDongleNumber - Retourne le numéro de série du dongle dans une chaîne.
- GetUpdateNumber - Retourne le prochain numéro de mise à jour.
- GetDataAreaSize - Retourne la taille des données.
- GetMaxAlgNum - Retourne le nombre maximum d'algorithmes détectés dans le dongle.
- GetExecs - Retourne le nombre d'exécutions restantes (-1 pour "illimité").
- GetExpDay - Retourne le jour d'expiration (-1 pour "illimité").
- GetExpMonth - Retourne le mois d'expiration (-1 pour "illimité").
- GetExpYear - Retourne l'année d'expiration (-1 pour "illimité").
- GetFeatures - Retourne la valeur du mot de fonction.
- GetNetUsers - Retourne le nombre maximum d'utilisateurs réseau simultanés autorisés pour ce dongle (-1 pour "illimité").
- GetAlgAnswer - Retourne la réponse de l'algorithme (seulement valide si ProtCheckWithAlg a été préalablement appelée avec succès).
- GetFDCapacity - Retourne la capacité en octets de la zone de données Flash Disk du dongle détecté. Seulement disponible pour les clés de type FD.
- GetFDDrive - Retourne l'unité locale attribuée à l'espace Flash Disk du dongle, exemple : f:\.
- GetDataRead - Retourne les données lues dans la zone de données sécurisée du dongle (seulement disponible si la fonction ReadString a été appelée préalablement avec succès).

3.3. Code exemple Visual Lisp

Remarque – Ce code exemple est uniquement destiné à illustrer la syntaxe d'appel des différentes méthodes et fonctions décrites précédemment. Vous devrez adapter ce code pour réaliser les appels nécessaires à la protection de votre programme, en fonction de vos besoins précis.

```

(vl-load-com) ; Vous devez IMPERATIVEMENT utiliser cette commande afin de charger les objets COM et
donc pouvoir faire appel à dpcom32.dll

(setq dinkeyobject (vlax-create-object "DPro.DPro.1")); Obtenir une référence à l'objet Dinkey Pro
(setq retvalue (vlax-invoke-method dinkeyobject "ProtCheck" 0 0 "")); Appel de la méthode 'ProtCheck' de
cet objet

(if (= retvalue 0) ; Test de la valeur de retour après test de la protection

(setq features (vlax-invoke-method dinkeyobject "GetFeatures")); Le test de protection s'est déroulé avec
succès lors de l'utilisation des fonctions Get pour lecture des paramètres du dongle

(exit) ; Si le test de protection a échoué, on sort. Dans votre code vous pourrez afficher les codes d'erreurs
accompagnés d'un message de votre choix.

)

(vlax-release-object dinkeyobject) ; Libération de l'objet Dinkey

```

3.4. Tableau des valeurs de FLAGS

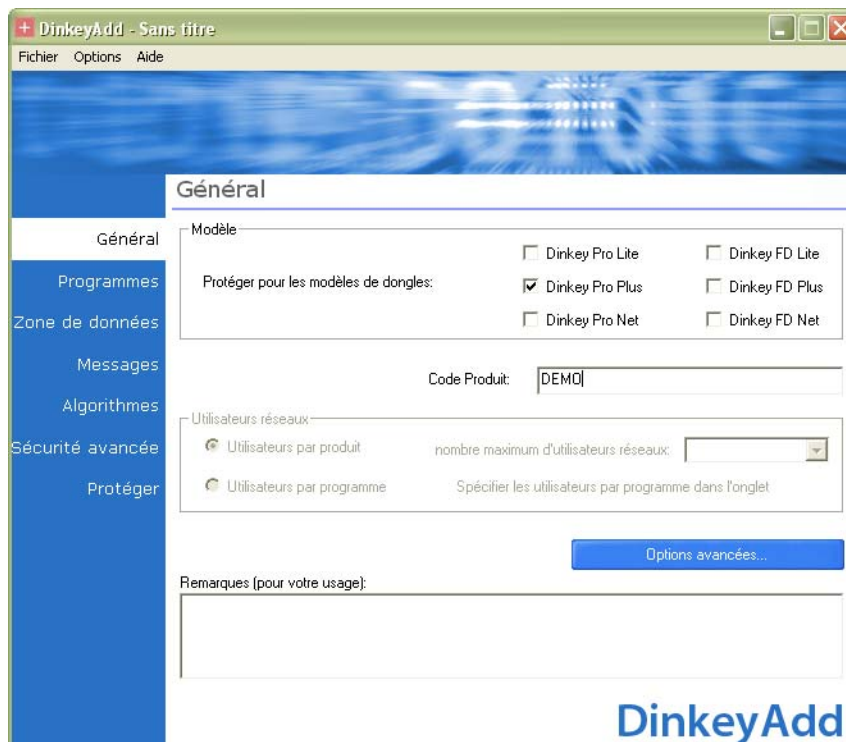
- 0 Test par défaut.
- 1 Décrémente de 1 le compteur d'exécutions.
- 2 Décrémente le compteur d'exécutions de la valeur indiquée dans `execs_decrement`. Utilisé avec les fonctions `DDSimpleCheckEx` et `DDSimpleCheckAlgEx`.
- 4 Démarre un utilisateur réseau.
- 8 Arrête un utilisateur réseau.
- 32 Vérifie les ports USB locaux avant de rechercher un dongle réseau (`DinkeyServer`).
- 64 Vérifier d'abord si un dongle réseau est disponible, puis scrute les ports USB locaux.
- 128 Fait un test de protection en vérifiant le nom du programme appelant indiqué dans `alt_prog_name`, avec les fonctions `DDSimpleCheckEx` et `DDSimpleCheckAlgEx`.
- 256 `DONT_SET_MAXDAYS_EXPIRY` Si la date d'expiration n'a pas été calculée, ne pas le faire cette fois.
- 512 `MATCH_DONGLE_NUMBER` Restreindre le test à un numéro de série de dongle précis (valeur indiquée dans le champ `DRIS`).

3.5. Protection du module `DPCOM32.DLL`

Ouvrez l'utilitaire `DinkeyAdd`.

Dans l'onglet "Général" :

- Précisez le type de clé que vous utilisez. En cas de doute, exécutez `DinkeyLook` afin d'afficher un diagnostic de la clé.
- Indiquez le Code Produit (pour les clés d'évaluation, celui-ci est "DEMO").



Dans l'onglet "Programmes" :

Détails de la protection des fichiers

Détails

Chemin source : C:\Program Files\WinRAR\dpwin32.dll

Chemin cible : C:\Program Files\WinRAR\protégé\dpwin32.dll

Programme appelant : [empty]

Méthode : API

Test régulier : pas de vérification régulière

Exécutions : illimité

Alerte exécutions : [empty]

Date d'expiration [jj/mm/aaaa] : illimité

Jours max : illimité

Alerte expiration : [empty]

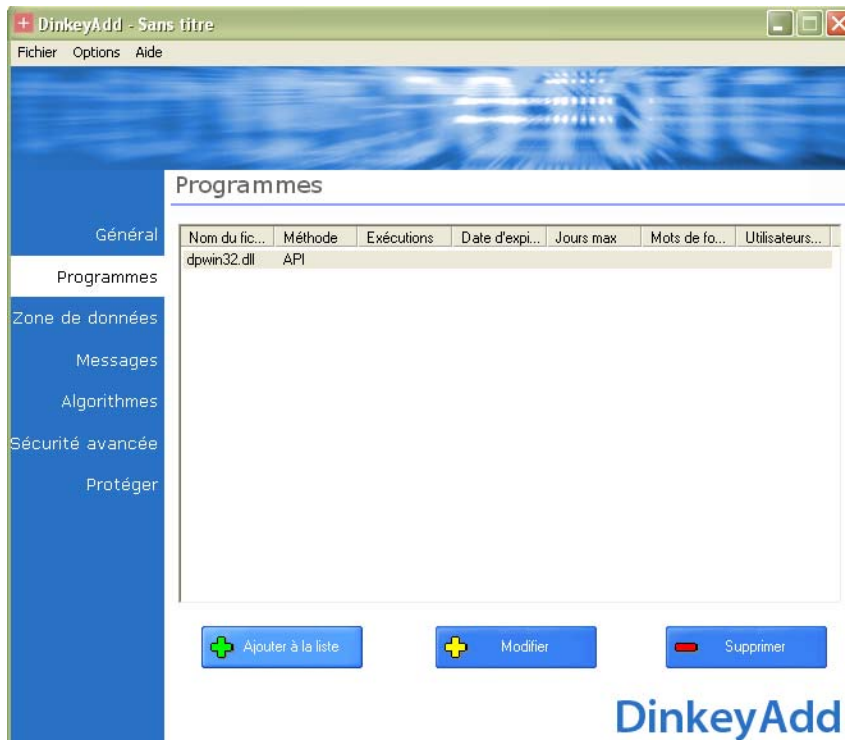
Mot de fonctions : 0

Utilisateurs réseaux : [empty]

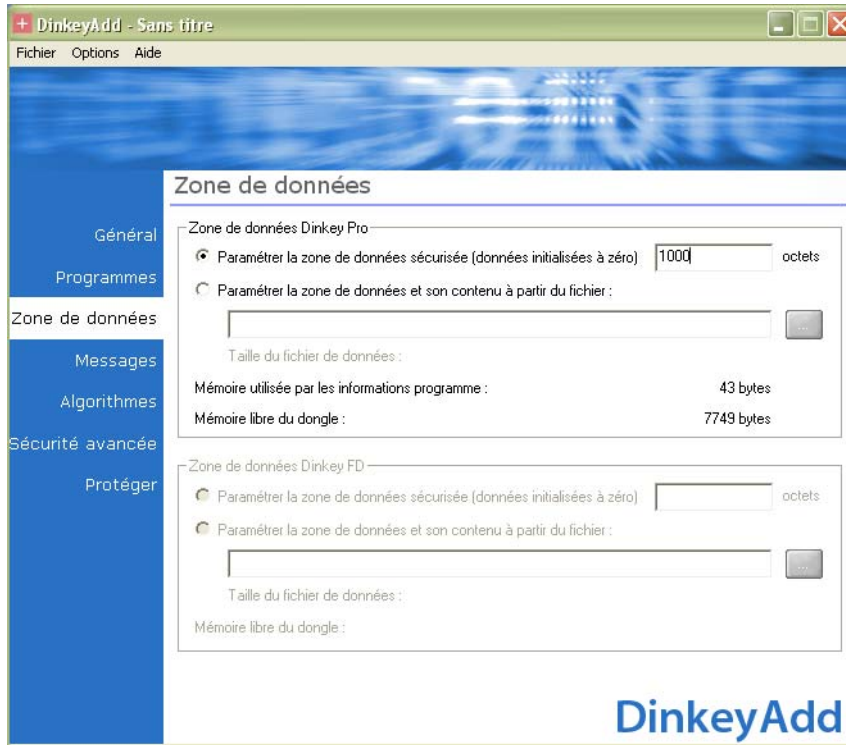
OK Annuler

- Cliquez sur le bouton "Ajouter à la liste".
- Dans le champ "Chemin source", pointez sur le fichier dpcom32.dll (celui-ci se trouve par défaut dans le sous-dossier "Modules" du dossier d'installation de DinkeyPRO).
- Dans le champ "Chemin cible", indiquez l'emplacement vers lequel vous souhaitez qu'une copie protégée de dpcom32.dll soit placée.
- Vérifiez que la méthode "API" est bien sélectionnée.
- Laissez les valeurs par défaut pour les champs suivants.
- Validez en cliquant sur le bouton OK.

Le fichier dpcom32.dll est ajouté à la liste des programmes.



Dans l'onglet "Zone de données" :

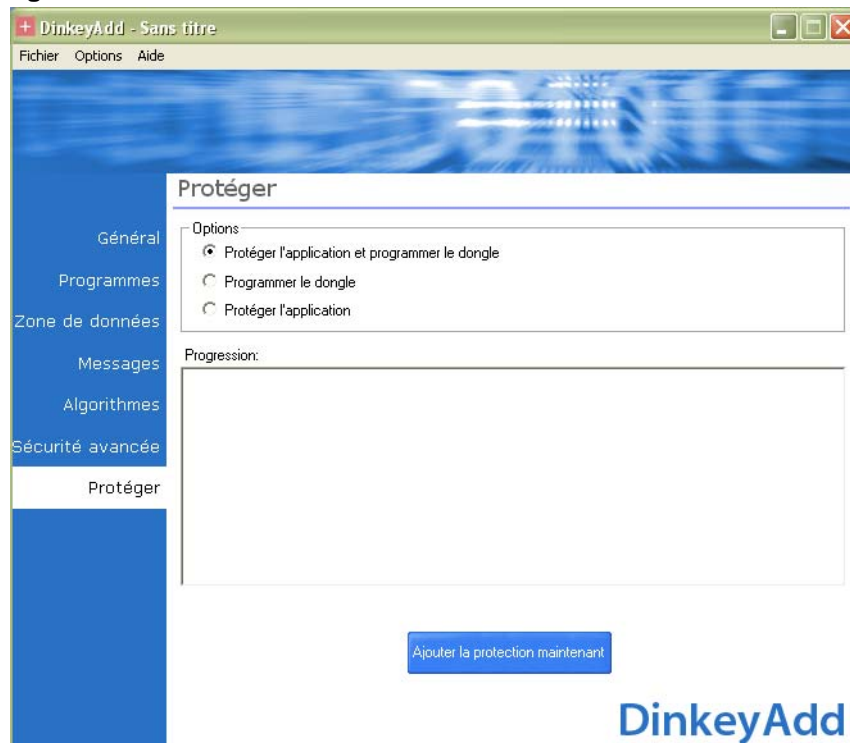


- Initialisez la taille de la zone de données sécurisée, par exemple 1000 octets. Le code exemple fourni permet de tester la lecture/écriture vers/à partir de la zone de données sécurisée. Ceci ne fonctionnera que si cette zone est initialisée avec une taille suffisante pour recevoir les données de test.

Remarques :

- Selon le type de clé (DinkeyPRO ou DinkeyFD) choisi dans l'onglet "Général", les paramètres de la zone de données correspondante seront activés dans l'onglet "Zone de données".
- Seules les versions Plus et Net disposent d'une zone de données sécurisées.

Dans l'onglet "Protéger" :



- Vérifiez que l'option "Protéger l'application et programmer le dongle" est sélectionnée.
- Assurez-vous que le dongle est connecté.
- Cliquez sur le bouton "Ajouter la protection maintenant"
- Un message de confirmation apparaît.

Vous disposerez alors d'un dongle correctement programmé, ainsi que d'une version protégée de `dpcom32.dll` dans le dossier vers lequel vous aurez pointé dans le champ "Chemin cible" de la fenêtre "Détails de la protection des fichiers".

4. Déploiement de votre application

Vous devrez fournir le fichier '`dpcom32.dll`' avec votre logiciel. Ce fichier doit être copié sur les postes utilisateurs et déclaré sous Windows reportez-vous à la section 2 de ce guide

Si '`dpcom32.dll`' n'est pas trouvé, une erreur surviendra.



aplika

*La Foltière - 37270 AZAY/CHER
Tél. 02 47 35 70 35 - Fax 02 47 35 70 25
e-mail : aplika@aplika.fr
www.aplika.fr*