

Démarrage rapide



DINKEY DONGLE

Protection de logiciels sous MAC avec DinkeyDongle



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 2.10 du 30/03/2010



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 DinkeyDongle MAC



1. Introduction	3
2. Le SDK DinkeyDongle pour MAC	3
3. Module objet, intégration et compilation	5
3.1. Module MAC DinkeyDongle :	5
3.2. Protection avec DDmac32.o	5
4. Premier test, interprétation des codes retour.....	6
5. Installation des drivers	6
5.1. Installation des drivers DinkeyDongle sous MAC OS X :	6
6. Fonctions de test de la clé des modules Objet DinkeyDongle	6
6.1. Précisions fonctions lecture/écriture	7
7. Codes Exemples	7
8. Variables de DinkeyDongle.....	7
9. Détail du paramètre Fonctions de ProtCheck.....	8

Guide d'intégration DinkeyDongle MAC

1. Introduction

Afin de vous aider dans les premiers pas de la pose de protection sur vos logiciels sous environnement MAC, vous trouverez dans ce guide de prise en main rapide l'essentiel des commandes et fonctionnalités pour la bonne utilisation de DinkeyDongle.

DinkeyDongle est compatible avec MAC OS X (Intel et Power PC)

Le SDK développeur est utilisable à partir de Mac OSX version 10.3 ou supérieur.

Les stations utilisateurs doivent être MAC OSX version 10.2 ou supérieur.

NOTE :

L'intégration de la protection DinkeyDongle à vos programmes MAC s'effectue en plusieurs phases :

- _ Appel des bibliothèques Dinkey pour effectuer des appels à la clé et vérifier son état
- _ Pose de la protection sur la ou les bibliothèques utilisées pour lier le programme protégé à vos clés
- _ Programmation des clés modèles Dinkey 2 et NET uniquement
- _ Installation sur les postes de vos clients.

2. Le SDK DinkeyDongle pour MAC

Tous les outils nécessaires à l'utilisation de DinkeyDongle sous MAC se situent dans le répertoire Mac du CD-ROM développeur.

Dans ce répertoire vous trouverez les packages pour l'utilisation de DinkeyDongle sous MAC. Vous trouverez également le fichier ddmac420.zip qui vous permettra l'installation des packages sur votre station Mac

Installation du SDK DinkeyDongle pour Mac OS X :

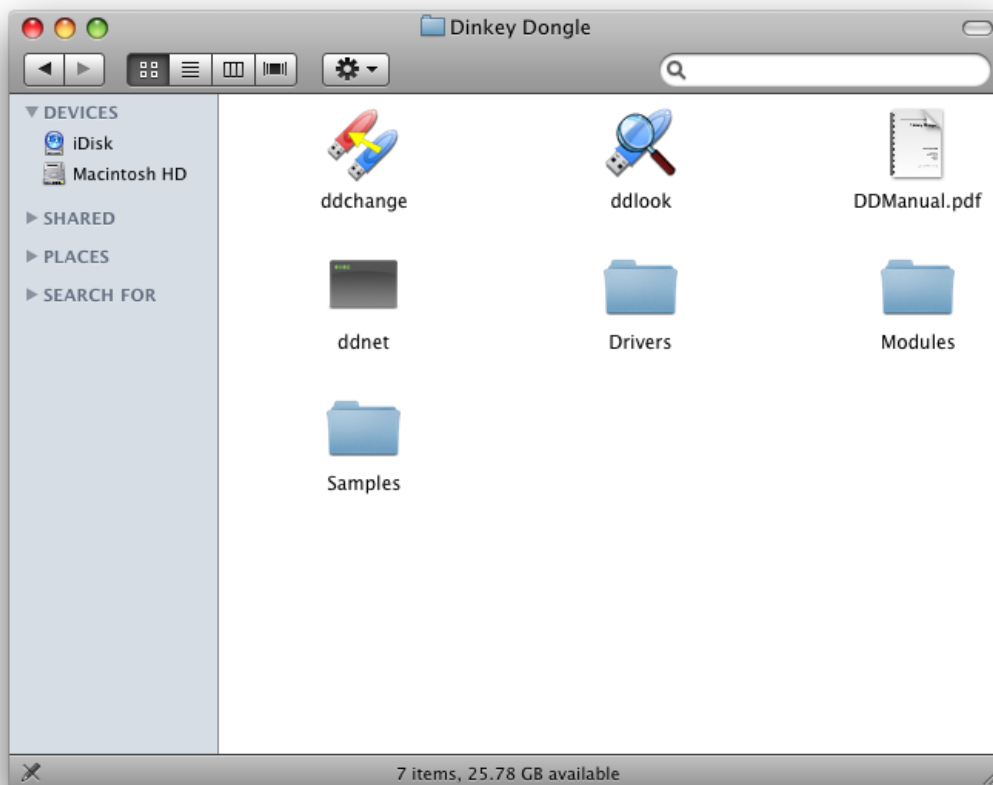
1. Décompressez le fichier ddmac420.zip file. Vous disposerez alors des deux fichiers suivants :
 - ddmac.pkg
 - ddUSBKeyDriver1.4.2.pkg.
2. Si le drivers DinkeyDongle n'est pas installé vous pouvez le faire en double cliquant maintenant sur ddUSBKeyDriver1.4.2.pkg. Après l'installation du pilote vous devez procéder au reboot de la machine.
3. Pour installer le SDK Mac de DinkeyDongle double cliquer sur ddmac.pkg.



Puis suivre les instructions d'installation.



Cliquez sur 'Continue' pour continuer l'installation du SDK. Une fois l'installation terminée vous avez alors à votre disposition un répertoire dans Applications/Dinkey Dongle



Le répertoire 'Drivers' contient le package permettant l'installation des pilotes Dinkey Dongle ddUSBKeyDriver1.4.2.pkg. Si vous ne l'avez pas déjà exécuté double-cliquez sur le package pour lancer l'installation du pilote. Après l'installation du pilote vous devez procéder au reboot de la machine.

C'est ce package que vous devrez fournir à vos clients pour l'installation des pilotes sur les stations MAC des utilisateurs.

Désinstallation du SDK DinkeyDongle pour Mac OS X :

Pour désinstaller le SDK DinkeyDongle :

- 1/ Supprimer le fichier "ddUSBKeyDriver.kext" situé dans le répertoire System/Library/Extensions.
- 2/ Supprimer le répertoire d'installation Dinkey Dongle Applications/Dinkey Dongle

3. Module objet, intégration et compilation

3.1. Module MAC DinkeyDongle :

Nous fournissons avec DinkeyDongle un module objet à utiliser pour établir le dialogue entre vos programmes et la clé DinkeyDongle :

DDmac32.o MAC OSX 10.2 et supérieur (MAC Intel et PowerPC)
DDmac32E.o MAC OSX 10.1 et supérieur (Power PC seulement) message en anglais uniquement
Dd32.dylib module Mac pour les programmes ne pouvant utiliser ddmac32.o (ex : RealBasic)
Dinkey4D.bundle module Mac pour les programmes 4D
Dinkey7.plugin module Mac pour les programmes FILEMAKER
LibDDINET.jnlib Authentification WEB pour poste client MAC

3.2. Protection avec DDmac32.o

Utiliser la librairie DDmac32.o de DinkeyDongle fournie dans le SDK MAC de DinkeyDongle (répertoire Mac du cd). L'objectif est de lier cette librairie à votre code afin de vérifier la présence de la clé et la validité de la licence.

1) Intégration vous devez ajouter les appels aux fonctions de la librairie DDmac32.o dans votre programme.

- Déclaration des variables (rcode, fonction, flags)
- Appel de la fonction (dd32)
- Test de la valeur de retour (si 0, le test est validé, sinon se référer à la liste des codes d'erreur fournie dans le manuel DinkeyDongle)

Pour plus d'informations sur les modes de protection DinkeyDongle n'hésitez pas à consulter le manuel DinkeyDongle. Vous pouvez également vous reporter au projet exemple en C/C++ ainsi qu'au code exemple du projet XCode, Pour CodeWarrior utilisé le code du projet en C.

2) Compilation pour la création de votre exécutable incluant les modules DinkeyDongle

Dans le cas le plus fréquent vous utiliser le Framework Carbon. Votre exécutable doit être compilé en ajoutant framework IOKit.framework à votre projet

3) Pose de protection pour lier le programme protégé avec vos clés DinkeyDongle

ATTENTION : La pose de protection et la programmation des clés doivent être réalisées avec le programme Windows DDADD.

Pour cela sous un poste Windows, installer le SDK Windows de DinkeyDongle, lancer le fichier setup.exe à partir du répertoire Windows, puis suivre les instructions d'installation.

Démarrer l'exécutable DDADD sur votre poste Windows

- Dans l'onglet *Général* cochez le type de clé puis saisissez un *code produit* de votre choix



- Dans l'onglet *Plus de détails* choisissez de *protéger le logiciel et le dongle* et *Protéger un programme avec un dongle portant mon SDSN*
- Dans l'onglet *Programmes* cliquez sur *Ajouter* puis sélectionnez le fichier à exécutable à protéger en méthode *objet*
- Dans l'onglet *Protéger* cliquez sur *Protéger*

Remarque :

_ Le fichier doit changer de taille après la pose de protection. Ce fichier deviendra alors votre fichier Master de protection à utiliser pour la distribution de vos logiciels.

_ Lors du premier appel à la protection le code erreur 71 est retourné si votre exécutable n'a jamais été protégé par DDADD.

_ DDMac32.o dispose de plusieurs fonctions décrites dans le chapitre 6 de ce guide.

4. Premier test, interprétation des codes retour

Une fois votre programme protégé avec DDADD vous obtiendrez un code de retour 0 en cas de succès et -28 en cas d'absence de la clé. Pour les autres valeurs du code de retour vous reporter au manuel du programmeur DinkeyDongle.

5. Installation des drivers

5.1. Installation des drivers DinkeyDongle sous MAC OS X :

Le pilote des clés Dinkey doit être installé sur la machine du poste client. Le driver doit être installé pour que le dongle soit reconnu.

a) Localiser le package ddUsbKeyDrivers1.4.2.pkg dans le repertoire mac à la racine du CD-ROM.

b) Double cliquer sur ce fichier pour installer les pilotes (Si vous avez déjà exécuté le package d'installation mac le pilote est déjà installer mais il faudra prévoir cette étape pour sur vos postes clients).

Note sur l'installation du pilote :

L'utilisation du package permet de copier le fichier ddUSBKeyDriver.kext dans le repertoire System | Library | Extensions folder.

Note sur la désinstallation du pilote :

Supprimer le fichier ddUSBKeyDriver.kext du repertoire System | Library | Extensions folder.

6. Fonctions de test de la clé des modules Objet DinkeyDongle

Les fonctions utilisables sont décrites ci-dessous :

1) int ProtCheck(int fonction, int flags);

Teste le dongle avec les valeurs spécifiées pour fonction en flags.

Remplir les variables globales de la structure DDMB (voir DDMB.h dans l'exemple en C)

ProtCheck () retournera 0 pour une interrogation de la protection avec succès ou retournera un code erreur.

ProtCheck () retournera 71 pour une interrogation de la protection avec succès mais avec l'utilisation d'un module non protégé avec DDADD.

2) int GetAlgVars(int[] vars);

Génère 4 entiers aléatoires à utiliser avec l'algorithme. Vous devez appeler cette fonction avant de réaliser un test de la protection si vous avez spécifié un algorithme.

3) int DDWrite(int flags, int offset, byte[] data);

Effectue une vérification de la protection et inscrit les données dans les offsets spécifiés de la zone de données du dongle. Initialise les variables du DDMB.

4) int DDRead(int fonction, int flags, int offset, int length, byte[] data);

Effectue une vérification de la protection avec les valeurs spécifiées pour fonction et flags. Lit les octets de données à partir de l'offset spécifié dans la zone de données.

Initialise les variables du DDMB.

5) String GetPCode()

Doit être effectué après l'une des fonctions 1-3. Retourne le Code Produit du dongle.

6) String GetPName()

Doit être effectué après l'une des fonctions 1-3. Retourne le Nom Produit stocké dans le dongle.

7) String GetSecureMsg()

Doit être effectué après l'une des fonctions 1-3. Retourne le message sécurisé.

6.1. Précisions fonctions lecture/écriture

1) Vous pouvez lire et écrire des données dans la zone de données sécurisée du dongle (seulement avec les Dinkey2/Net) mais vous pouvez seulement lire/écrire des chaînes de caractères, pas des données brutes (fichiers).

7. Codes Exemples

Les deux exemples C/C++ et le projet XCode fournis proposent différents appels aux fonctions.

Vous les trouverez sur le CD-ROM Dinkey dans le répertoire Mac ils seront accessibles après l'installation du package.

Le projet Xcode et son source en C est valable pour codewarrior.

Nous considérons que vous disposez déjà du Framework Carbon installé. Vous devrez ajouter IOKit.Framework à votre projet

8. Variables de DinkeyDongle

dongle_no	Numéro de série du dongle.
sdsn	SDSN stocké dans le dongle.
prodcode	Code produit stocké dans le dongle.
execs	Nombre d'exécutions autorisées par le dongle (-1 pour illimité).
exp_day	Jour de la date d'expiration (31/15/2107 pour illimité).
exp_month	Mois de la date d'expiration (31/15/2107 pour illimité).
exp_year	Année de la date d'expiration (31/15/2107 pour illimité).
features	Mot de fonction stocké dans le dongle.
update	Numéro de mise à jour actuel du dongle stocké dans la clé
maxnetusers	Nombre maximum d'utilisateurs réseaux simultanés autorisé.
usb	Type de dongle : 1 - port parallèle, 2 - USB.
data	Valeur de la chaîne de caractères contenue dans la zone de données.
model	Modèle de dongle : 1 Dinkey 1, 2 - Dinkey 1S, 3 - Dinkey 2, 4 - DinkeyNet.

Remarque : Toutes les variables du dongle sont stockées dans une structure nommée DDMB. La description complète du bloc mémoire DDMB est disponible dans le manuel du développeur 'Structure DinkeyDongle' référence 11-1. Pour une recherche plus facile dans le PDF du manuel pensez à utiliser les signets.

9. Détail du paramètre Fonctions de ProtCheck

Le champ fonction correspond au numéro de fonction utilisé pour vérifier la protection.
Plus de détails sur les fonctions :

Le champ fonction doit être renseigné. Il peut avoir les valeurs suivantes :

Fonction = 0. Cette fonction permet la mise à jour des paramètres du dongle et effectue la vérification de la protection (décompte d'un droit d'exécution). Si vous utilisez DinkeyNet, cette fonction permet le démarrage d'un utilisateur réseau et vérifie également si le nombre maximum d'utilisateurs réseau simultanés n'est pas dépassé.

Fonction = 1. Cette fonction vérifie la protection et permet d'écrire dans la zone de donnée du dongle. Vous devez initialiser les champs `rw_length`, `rw_offset` et `rw_data_ptr` fields pour écrire dans la zone de donnée (voir ci-dessous).

Fonction = 2. Cette fonction teste le dongle sans mettre à jour les paramètres et sans démarrer d'utilisateur réseau.

Fonction = 3. Cette fonction teste la protection et permet la mise à jour des paramètres du dongle mais ne démarre pas d'utilisateur réseau.

Fonction = 4. Cette fonction teste la protection, démarre un utilisateur réseau mais ne met pas à jour les paramètres du dongle.

Fonction = 6. Cette fonction permet de forcer la fermeture d'un utilisateur réseau. Elle peut être utilisée dans certains cas où la fermeture automatique de l'utilisateur réseau poserait problème, cela est très rare. Ou dans le cas où vous souhaitez fermer l'utilisateur réseau du dongle avant de fermer votre logiciel. Pour utiliser cette fonction, utiliser la routine DDWIN32 avec fonction initialisée à 6. Cela ne teste pas la protection.

Si vous devez vérifier la protection plusieurs fois, nous vous conseillons d'utiliser la fonction 0 la première fois puis d'appeler la fonction 1 ou 2. Si vous appelez plusieurs fois la fonction 0 cela décomptera une utilisation à chaque fois (ou un utilisateur réseau).

DinkeyNet libère automatiquement l'utilisateur réseau lorsque l'application est fermée. Cependant, il peut arriver qu'il soit nécessaire de forcer la libération de l'utilisateur réseau. Pour cela appelez la routine `_dd` avec la fonction 6.

Vous pourrez bien sûr appeler plusieurs fois les fonctions de DinkeyDongle.



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