



Patrice A. BONNEFOY
MVP Windows® Client
www.vista-system.eu
Patrice@vista-system.eu

Windows® Vista™

Présentation de Boot Configuration Data (BCD)



*Article réalisé avec la version beta de Windows® Vista™ Ultimate build 5600 .
Cet article a fait l'objet d'une publication par IT Media, revue IT Pro Magazine de
décembre 2006, page 30.*

INTRODUCTION.....	2
EFI - EXTENSIBLE FIRMWARE INTERFACE : SERA OU NE SERA PAS ?.....	3
NOUVELLE ARCHITECTURE DE DEMARRAGE POUR WINDOWS VISTA™	3
BOOT CONFIGURATION DATA EN DETAIL.....	4
BCD STORES.....	4
BCD OBJECTS.....	4
APPLICATION OBJECTS.....	4
INHERITABLE OBJECTS.....	4
DEVICE OBJECTS.....	5
BCD ELEMENTS.....	5
LES OUTILS POUR BOOT CONFIGURATION DATA.....	5
SYNTAXE DE COMMANDES AVEC BCDEDIT.EXE.....	5
CONCLUSION.....	6
POUR ALLER PLUS LOIN.....	6

Introduction

Avec l'arrivée de Windows® Vista™, Microsoft® introduit une nouvelle gestion de l'environnement de démarrage. Cette nouveauté permettra de mieux gérer la diversité des nouveaux logiciels et matériels actuels. Ce changement notable du lancement des OS Windows® de nouvelle génération s'exécute dans un environnement encore plus sécurisé.

L'évolution de cet environnement ajoute un nouvel outil de gestion du processus d'initialisation (BCDEdit.exe). Cet outil permet la configuration et le contrôle des données d'initialisation (Boot Configuration Data) utilisées pour le lancement des systèmes Longhorn.

Nous allons voir dans cet article comment Windows® Vista™ est démarré puis, comment manipuler les données d'initialisation contenues dans cette BCD.

Présentation

Lorsqu'un ordinateur personnel est démarré, le processus est différent selon que l'initialisation se passe depuis un lecteur de disque, une clé USB, un lecteur réseau, etc. Mais dans tous les cas, c'est le BIOS qui intervient dans la lecture du Master Boot Record (MBR). Il recherche la partition active, charge le secteur de boot en mémoire; celui-ci transfère la commande au chargeur Windows® NTLDR qui s'occupe de faire basculer le processeur du mode réel vers le mode linéaire 32 bits.

NTLDR charge, depuis la racine de la partition système, le système sélectionné dans le fichier texte Boot.ini, exécute NTDetect.com qui se charge de reconnaître tous les matériels disponibles raccordés à la machine et enfin, charge le noyau puis, la couche d'abstraction matérielle et enfin la couche SYSTEM pour y loger les périphériques détectés. (*La détection matérielle est complète depuis Windows® 2000, système totalement PnP*). Chaque système à démarrer trouve donc ses informations de lancement dans ce légendaire fichier Boot.ini.

Pour les machines à noyau « NT », cela fonctionne de cette manière depuis Windows® NT 3.1 jusqu'à maintenant où Microsoft® adapte ses systèmes d'exploitation aux conditions actuelles. En même temps Microsoft® poursuit avec Windows® Vista™ sa politique de qualité et de sécurité de ses systèmes.

EFI - Extensible Firmware Interface : sera ou ne sera pas ?

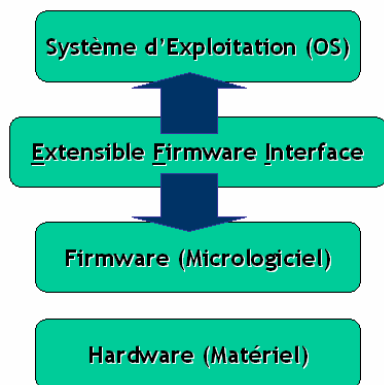


Image 1

Depuis plusieurs années, un forum de sociétés majeures de l'industrie informatique (Intel® et Microsoft® membres fondateurs) travaillent conjointement à supplanter ou à remplacer le BIOS par l'EFI. BIOS, trop vieux, pas du tout évolutif, datant du début des années 70 - ou depuis 1981 si l'on considère seulement l'IBM PC - ne peut plus répondre aux exigences complexes des matériels et logiciels actuels. EFI traitera les fonctions d'initialisation ainsi que les services d'exécution des OS installés.

EFI est une interface (écrite en langage C donc, très évolutive) entre le matériel de la machine et le(s) système(s) d'exploitation installé(s) (image 1). EFI sera compatible avec le Bios pour démarrer les systèmes non compatibles.

Actuellement, EFI n'est pas encore finalisé pour tous les acteurs et, finalement, ne sera pas supporté par Windows® Vista™ (*annonce Microsoft de mars 2006 dernier lors de l'Intel Developer Forum de San Francisco*) à sa commercialisation. Seuls, au moment de leurs sorties, les systèmes d'exploitation Longhorn Server x64 et IA64 seront compatibles EFI. Cependant, on peut le regretter, Windows® Vista™ sera prêt pour cette technologie seulement avec le Service Pack 2 ; ce qui nous emmène en 2008.

Nous retiendrons aujourd'hui que le démarrage d'une machine EFI est complètement différent de ce que nous connaissons. Cet environnement n'étant pas d'actualité pour Windows® Vista™, cet article est traité dans un environnement de compatibilité BIOS.

Nouvelle architecture de démarrage pour Windows® Vista™

Windows® Vista™ présente une nouvelle architecture d'initialisation de démarrage, de configuration et de stockage des données appelé BCD (Boot Configuration Data).

Le vieillissant NTLDR est maintenant remplacé par deux composants :

- Windows Boot Manager (*Bootmgr.exe*)
- System specific boot loaders (*Winload.exe* et *Winresume.exe*).

Bootmgr n'est que générique et non lié à un système d'exploitation tandis que Winload et Winresume sont optimisés pour le système qu'ils démarrent. Ces fichiers résident à la racine de chaque version de Windows®.

Bootmgr puise les informations de démarrage dans la BCD et affiche le menu de sélection des systèmes à démarrer.

L'utilisateur choisit son OS et, à cet instant, le chargeur Winload crée l'environnement de démarrage du système concerné, charge le noyau (KERNEL) de Windows® Vista™, la couche d'abstraction matérielle (HAL) et enfin les divers pilotes de périphériques.

Winresume est le chargeur d'état qui restore le contenu complet d'une session à sa sortie du mode hibernation.

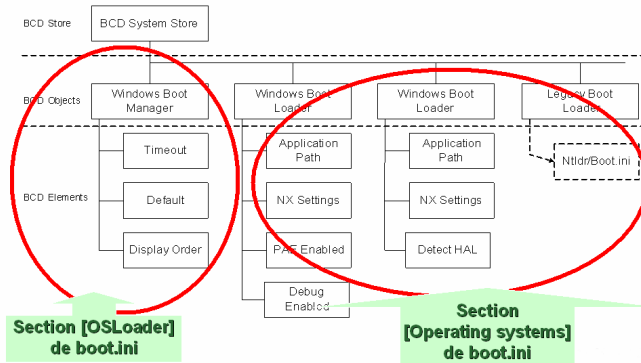
Dans le cas d'un environnement de démarrage multiple contenant une ou plusieurs versions précédentes de Windows®, le chargement est un peu différent.

Bootmgr charge NTLDR, lui-même affiche alors le menu de démarrage issu du fichier texte boot.ini (Avec EFI, ces données se trouvent stockées dans la NVRam).

Boot Configuration Data en détail

Jugé trop vulnérable aux attaques malveillantes par Microsoft®, le fichier

texte Boot.ini contenant les paramètres de démarrage des systèmes Windows® actuels est supprimé. Windows® Vista™ ainsi que les versions suivantes puisent leurs informations de démarrage dans une clé de la base de registres appelée BCD située dans la ruche HKLM : HKEY_LOCAL_MACHINE/BCD00000000. Au niveau de la sécurité, un Administrateur peut appliquer une stratégie sur cette clé de registre ; ce qui en fait un atout indéniable vis à vis des droits d'initialisation pour chaque utilisateur. BCD



est administrable à distance donnant ainsi la possibilité de corriger ou dépanner une machine.

Il est très facile de manipuler ce magasin de données puisqu'il est situé dans une architecture bien connue des Administrateurs et Développeurs : la base de registres.

Ce magasin de données est architecturé suivant une hiérarchie de 3 composants : Le magasin lui-même ou BCD Stores, les objets ou BCD Objects ainsi que les éléments de mises au point ou BCD elements. L'image 2 donne un aperçu de l'organisation de la nouvelle architecture en comparaison du fichier boot.ini.

BCD Stores

Il contient les objets et les éléments ; Ce magasin comprend au moins 2 objets : le Windows® Boot Manager et le Windows® Boot Loader.

BCD Objects

Il contient les éléments sous forme de GUID décrivant généralement des options d'initialisation pour les chargeurs de Windows® Vista™. On y trouve 3 catégories d'objets :

- Application Objects,
- Inheritable Objects,
- Device Objects.

Application Objects

- Firmware Boot Manager est le chargeur des systèmes EFI,
- Windows® Boot Manager contient les paramètres d'initialisation du système à charger par défaut, c'est-à-dire le Time out et le système à démarrer,
- Windows® Boot Loader contient les lignes formant le menu des systèmes à démarrer paramétrées des différentes options de lancement,
- Legacy Boot Loader ou Windows® ntlldr contient le menu des versions précédentes de Windows®.
- Windows® Resume Loader contient les informations d'hibernation des systèmes,

Inheritable Objects

Ces objets permettent la gestion de la mémoire, les mises au point du système...

- Windows® Memory Tester.

Device Objects

Ces méthodes permettent de renseigner la plupart des éléments complexes comme, par exemple, une image WIM pour laquelle on aurait stocké sa localisation ainsi que le port réseau en cas de démarrage par celui-ci.

BCD Elements

Les éléments servent généralement à la mise au point, aux paramètres de mise au point des noyaux systèmes, au stockage des paramètres de lancement (boot.ini) des versions précédentes de Windows®.

Le fichier Boot.ini contenait ces informations de manière globale. Avec BCD, chaque élément de description de démarrage d'un objet est contenu dans un élément séparé les rendant ainsi facilement exploitables.

Les outils pour Boot Configuration Data

Les données de cette BCD ne sont pas accessibles directement ; elles y sont stockées de manière binaire. Elles ne peuvent donc être modifiées que par les outils qui ont été prévus pour cela.

- On peut modifier sommairement l'environnement de démarrage depuis le panneau de configuration par les fonctions avancées d'Administration du système. L'Administrateur peut y modifier l'élément « Time out » ainsi que le système à démarrer par défaut.

- On peut modifier l'environnement de démarrage avec l'outil MSConfig.exe.

- La nouveauté pour gérer cette BCD est : BCDEdit.exe.

BCDEdit ne possède pas d'interface graphique (pour le moment ?) ; il s'utilise donc dans une invite de commande. Il s'agit du seul outil complet Microsoft® permettant d'entreprendre toutes les opérations de modification et de création de l'environnement de démarrage. Par contre, pour les modifications de types de données complexes, les développeurs préféreront les BCD WMI API pour manipuler ces informations ou en créer de nouvelles.

L'outil BCDEdit se trouve dans le dossier %WINDIR%\System32 ; il n'est directement accessible que par le compte d'Administrateur principal de la machine ou par élévation des droits d'administration pour les autres comptes. Il n'est d'ailleurs pas conseillé de modifier les droits utilisateurs sur cet outil.

Syntaxe de commandes avec BCDEdit.exe

Voyons maintenant la syntaxe des commandes lancées à partir d'une invite de commandes en tant qu'Administrateur :

BCDEdit.exe /Command [Paramètre1] [Paramètre2]...

BCDEdit / ? affiche les commandes disponibles.

BCDEdit / ? command où command est le nom de la commande dont on désire connaître les paramètres associés.

BCDEdit /enum ou /v donne la liste des entrées contenues dans la BCD.

BCDEdit /default indique quel système est démarré lorsque le temps d'affichage du menu expire.

BCDEdit /timeout spécifie en secondes le temps au bout duquel le système par défaut est démarré.

Création d'une entrée :

BCDEdit /copy {GUIDàcopier} /d "Ma Description" crée une entrée de lancement d'un nouvel OS avec pour description "Ma Description". Le nouveau GUID est automatiquement créé,

BCDEdit /set {nouveauGUID} device partition=f :

BCDEdit /set {nouveauGUID} osdevice partition=f : affecte la nouvelle entrée à l'OS situé sur la partition f :

BCDEdit /displayorder {nouveauGUID} -addlast rend disponible cette entrée et la l'ajoute à la fin de la liste du menu de démarrage.

Exemple concret de création :

```
C:\Users\Administrator> BCDEdit /copy {24a500f2-12ea-11db-a536-b7db70c06ac2} /d "Build 5600
French"
The entry was successfully copied to {88f877d4-2b8b-11db-a7ac-00a0c9e528a8}.
C:\Users\Administrator> BCDEdit /set {88f877d4-2b8b-11db-a7ac-00a0c9e528a8} device partition=f:
The operation completed successfully.
C:\Users\Administrator> BCDEdit /displayorder {88f877d4-2b8b-11db-a7ac-00a0c9e528
a8} /addlast
The operation completed successfully.
```

BCDEdit /displayorder {GUID1} {GUID2} {GUID3} spécifie l'ordre dans lequel les entrées sont affichées dans le menu de démarrage.

BCDEdit /displayorder {466f5a88-0af2-4f76-9038-095b170dc21c} -addlast pour afficher, à la fin de la liste du menu, la ligne de démarrage d'une version précédente de Windows®. Il est à noter que ce GUID est prédéfini pour l'entrée {ntldr}.

BCDEdit /delete {GUID} supprime une entrée du menu. Spécifier le paramètre -f permet de la supprimer totalement de la BCD mais de perdre ainsi une entrée peut-être valide.

BCDEdit /bootsequence {GUID1} {...} {...} indique le système d'exploitation à démarrer au prochain démarrage de la machine ou l'ordre dans lequel la liste des objets du menu s'affiche. A la suite de ce lancement, le menu reprend l'ordre précédent.

BCDEdit /set {dbgsettings} {GUID} spécifie des paramètres de mise au point pour une entrée.

BCDEdit /debug {cbd971bf-b7b8-4885-951a-fa03044f5d71} spécifie un démarrage de mise au point pour le système correspondant à ce GUID.

Conclusion

Cette nouvelle architecture représente une avancée majeure pour les Administrateurs et Développeurs qui voient par-là une machine mieux sécurisée et adaptée aux évolutions futures.

Cette architecture est organisée dans un contexte bien connu : la base de registres. Les principaux utilisateurs n'en seront pas déroutés.

Pour aller plus loin

Le consortium UEFI : <http://www.uefi.org>

(Articles en anglais)

Introduction aux options de démarrage : http://msdn.microsoft.com/library/en-us/DevTest_g/hh/DevTest_g/BootIni_08d86912-57de-439d-b3d5-2a3ad7b8ad17.xml.asp

Boot Configuration Data : <http://msdn.microsoft.com/library/en-us/BCD/bcd/portal.asp>

BCD pour développeurs : http://msdn.microsoft.com/library/en-us/BCD/bcd/bcd_reference.asp

Environnement BCD avec Windows® Vista :

<http://www.microsoft.com/whdc/system/platform/firmware/bcd.mspx>