

# L'USB en bref

## *Donner un sens au standard USB*

### **Les Descripteurs USB**

Tous les appareils USB ont une hiérarchie de descripteurs qui détaillent pour le compte de l'hôte des informations l'instruisant sur la nature de l'appareil, qui l'a réalisé, quelle version USB il supporte, de combien de manières il peut être configuré, le nombre de terminaisons et leurs types etc...

Les descripteurs les plus courants sont :

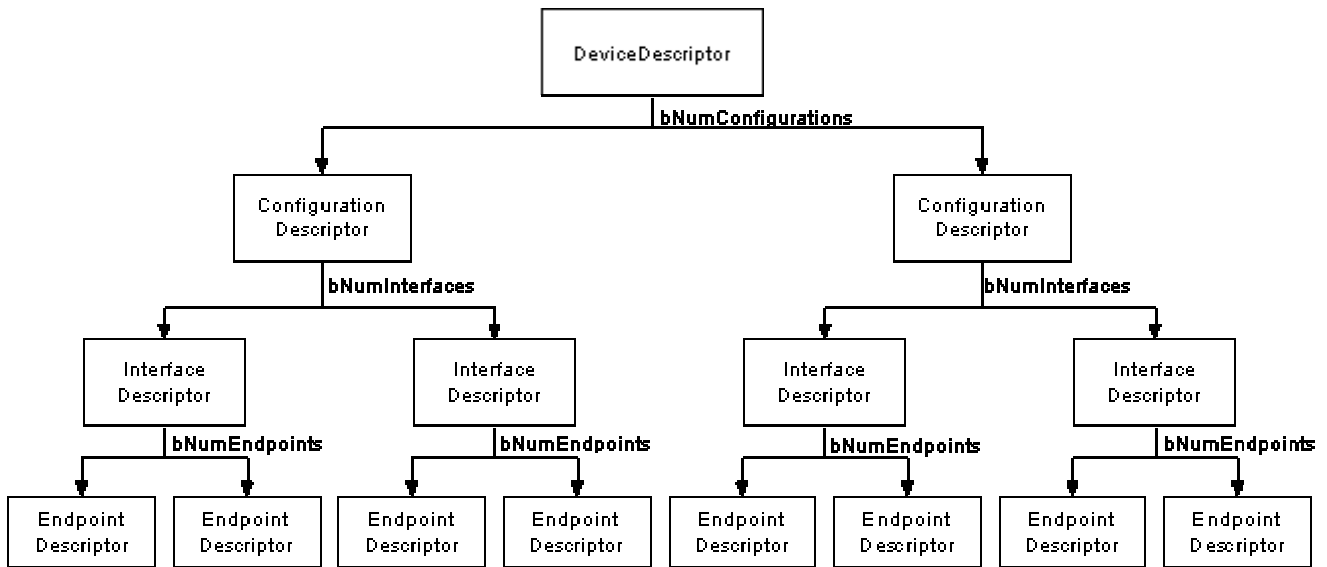
- **Descripteurs d'Appareils**
- **Descripteurs de Configurations**
- **Descripteurs d'Interfaces**
- **Descripteurs de Terminaisons**
- **Descripteurs de Chaînes**

Les appareils USB ne peuvent avoir qu'un seul descripteur d'appareil. Le descripteur d'appareil inclut des informations qui précisent la révision USB à laquelle l'appareil se soumet, les IDs ( Identificateurs d'Appareils) du produit et du constructeur utilisés pour charger les pilotes logiciels appropriés et le nombre possible de configurations que l'appareil peut avoir. Le nombre de configurations indique combien de ramifications de descripteurs de configurations sont appelées à suivre.

Le descripteur de configuration précise des valeurs comme la quantité de puissance qu'utilise cette configuration particulière, si l'appareil est auto-alimenté ou alimenté par le bus et le nombre d'interfaces qu'il possède. Quand un appareil est énuméré, l'hôte lit les descripteurs d'appareils et peut décider de la configuration à valider. Il peut seulement valider une configuration à la fois.

Par exemple, il est possible d'avoir une configuration d'alimentation de bus de grande puissance et une configuration auto-alimentée. Si l'appareil est branché à un hôte possédant une alimentation électrique secteur, le pilote logiciel de l'appareil choisira peut-être de permettre la configuration d'alimentation du bus de grande puissance tolérant ainsi que l'appareil soit alimentée sans être relié au secteur, cependant s'il est connecté à un laptop (ordinateur portable) ou à un organisateur personnel, il pourra valider la seconde configuration ( auto-alimentée) exigeant de l'utilisateur de brancher son appareil sur un point d'alimentation (secteur).

Les paramètres de configurations ne sont pas limités aux différences d'alimentations. Chaque configuration pourrait être alimentée de la même façon et drainer le même courant, et avoir cependant des combinaisons de terminaisons et d'interfaces différentes. Toutefois il faut tenir compte que la modification de la configuration exige l'arrêt de toute activité sur chaque terminaison. Tandis que l'USB propose cette flexibilité, très peu d'appareils possèdent plus d'une configuration.

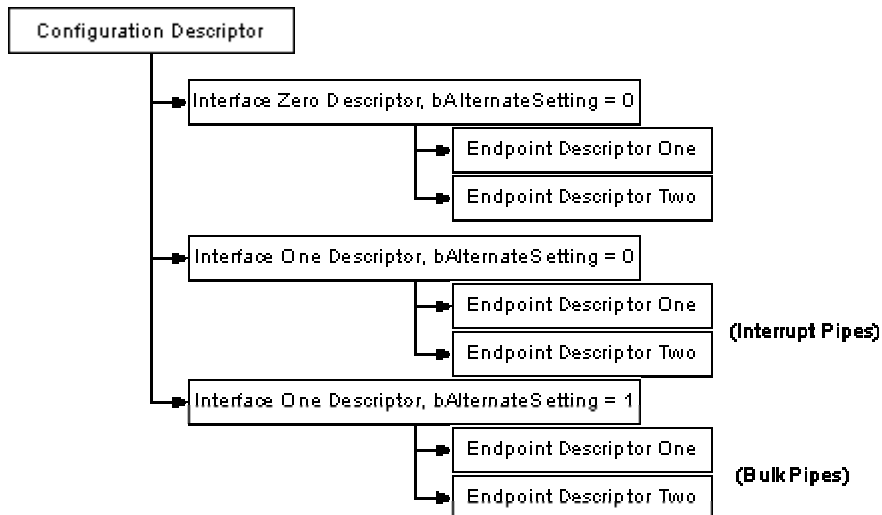


Le descripteur d'interface peut être vu comme un "en tête" ou un regroupement de terminaison à l'intérieur d'un groupe fonctionnel accomplissant une seule fonctionnalité de l'appareil. Par exemple, vous pouvez avoir un appareil multifonctions : fax / scanner / imprimante. Le descripteur d'interface 1 pourra décrire les terminaisons de la fonction fax, le descripteur d'interface 2, la fonction scanner et le descripteur d'interface 3 la fonction imprimante. Contrairement au descripteur de configuration, il n'y a pas de limitations à avoir une seule interface validée à la fois. Un appareil peut avoir un ou plusieurs descripteurs d'interfaces validées en même temps.

Les descripteurs d'interfaces ont un champ **bInterfaceNumber** précisant le numéro de l'interface et un **bAlternateSetting** qui autorise l'interface à modifier ses paramètres au vol. Par exemple on peut avoir un appareil avec 2 interfaces, interface 1 et interface 2. Interface 1 a **bInterfaceNumber** mis à 0 indiquant qu'il est le premier descripteur d'interface et un **bAlternativeSetting** de 0.

L'interface 2 aura un **bInterfaceNumber** mis à 1 indiquant qu'il est la seconde interface et un **bAlternativeSetting** de 0 (par défaut). On pourra donc y faire entrer un autre descripteur, comprenant lui aussi un **bInterfaceNumber** mis à 1 indiquant qu'il est la seconde interface, mais cette fois le **bAlternativeSetting** mis à 1, indiquant que ce descripteur d'interface peut représenter un paramètre alternatif à celui de l'autre descripteur d'interface 2.

Quand cette configuration est validée, les 2 premiers descripteurs d'interfaces avec **bAlternativeSettings** égal à 0 sont utilisés. Toutefois pendant le fonctionnement, l'hôte peut envoyer une demande imposée SetInterface ( Sélection d'Interface) à l'interface 1 avec un alternative setting (paramètre alternatif) à 1 pour valider l'autre descripteur d'interface.



Cela est plus avantageux que d'avoir 2 configurations, dans le sens où l'on peut transmettre des données via l'interface 0 tandis que l'on change les paramètres de terminaisons associés à l'interface 1 sans affecter l'interface 0.

Chaque descripteur de terminaison est utilisé pour spécifier le type de transfert, la direction, l'intervalle d'interrogation et la taille maximale de paquet pour chaque terminaison. La terminaison 0, la terminaison de commandes par défaut est toujours supposée être une terminaison de commandes et en tant que tel ne possède jamais de descripteur.

### Composition des descripteurs USB

Tous les descripteurs relèvent d'un schéma commun. Le premier octet précise la longueur du descripteur, tandis que le second octet indique le type de descripteur. Si la longueur du descripteur est plus petit que ce que définit la spécification, alors l'hôte doit l'ignorer. Toutefois si la taille est plus grande que prévue, l'hôte ignorera les octets supplémentaires et ne commencera à rechercher le prochain descripteur qu'à la fin d celui-ci.

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du descripteur en octets
1	bDescriptionType	1	Constante	Type de descripteur
2	...	n		Début des paramètres destinés au descripteur

### Descripteurs d'appareils

Le descripteur d'appareil d'un appareil USB représente l'appareil en entier. En conséquence un appareil USB ne peut avoir qu'un seul descripteur d'appareil. Il donne des informations élémentaires et pourtant fondamentales sur l'appareil telles la version USB supporté, la taille maximale de paquet, les IDs constructeur et produits et le nombre de configurations possibles que peut avoir l'appareil. Le format du descripteur d'appareil est montré ci-dessous :

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du descripteur en octets(12 octets)
1	bDescriptorType	1	Constante	Descripteur d'Appareil (0x01)
2	bcdUSB	2	BCD	Numéro de spécification USB auquel l'appareil doit aussi se soumettre.
4	bDeviceClass	1	Classe	Code classe (class) (Assigné par USB org)  si égal à 0, chaque interface précise son propre code classe  si égal à 0xFF, le code classe est précisé par le constructeur.  Autrement le champ est un code classe valable.
5	bDeviceSubClass	1	Sous Classe	Code sous Classe (Assigné par USB org)
6	bDeviceProtocol	1	Protocole	Code Protocole (Assigné par USB Org)
7	bMaxPacketSize	1	Nombre	Taille maximale de paquet pour la terminaison Zéro. Les tailles conformes sont 8, 16, 32, 64
8	idVendor	2	ID	IDentification du fournisseur(Assigné par USB Org)
10	idProduct	2	ID	IDentification du produit (Assigné par le Fabricant)
12	bcdDevice	2	BCD	Numéro de version de l'appareil
14	iManufacturer	1	Index	Index de descripteur de chaîne du fabricant
15	iProduct	1	Index	Index de descripteur de chaîne du produit
16	iSerialNumber	1	Index	Index de descripteur de chaîne du numéro de série
17	bNumConfigurations	1	Entier	Nombre de configurations possible

Le champ **bcdUSB** rapporte la version USB la plus haute que peut supporter l'appareil. La valeur est en binaire codé décimal avec un format de 0xJJMM ou JJ est le numéro de version de poids fort, M le numéro de version de poids faible et N correspond au numéro de sous-version c'est à dire USB 2.0 est inscrit comme 0x0200, USB 1.1 comme 0x0110 et USB 1.0 comme 0x0100.

Les **bDeviceClass**, **bDeviceSubClass** et **bDeviceProtocol** sont utilisés par le système d'exploitation pour trouver un pilote logiciel de classe pour votre appareil. Habituellement seul le **bdeviceclass** est positionné au niveau de l'appareil. La plupart des spécifications de classe choisissent de s'identifier au niveau de l'interface et en conséquence positionnent le **bdeviceclass** à 0x00. Cela permet à un appareil de supporter plusieurs classes.

Le champ **bMaxPacketSize** rapporte la taille maximale pour la terminaison zéro. Tous les appareils doivent supporter la terminaison zéro.

Le champ **idVendor** et **idProduct** sont utilisés par le système d'exploitation pour trouver un pilote logiciel pour votre appareil. L'identification constructeur (vendor ID) est assignée par USB-IF.

Le champ **bcdDevice** a le même format que **bcdUSB** et est utilisé pour fournir un numéro de version d'appareil. Cette valeur est assignée par le développeur.

3 descripteurs de chaînes de caractères existent pour fournir des détails du fabricant, du produit et du numéro de série. Il n'y a pas d'obligation à avoir des descripteurs de chaînes de caractères. Si aucun descripteur de chaînes de caractères n'est présent, il faudrait utiliser un index de zéro.

**bNumConfigurations** définit le nombre de configuration que l'appareil peut supporter et sa vitesse normale d'exécution.

### Les Descripteurs de Configurations

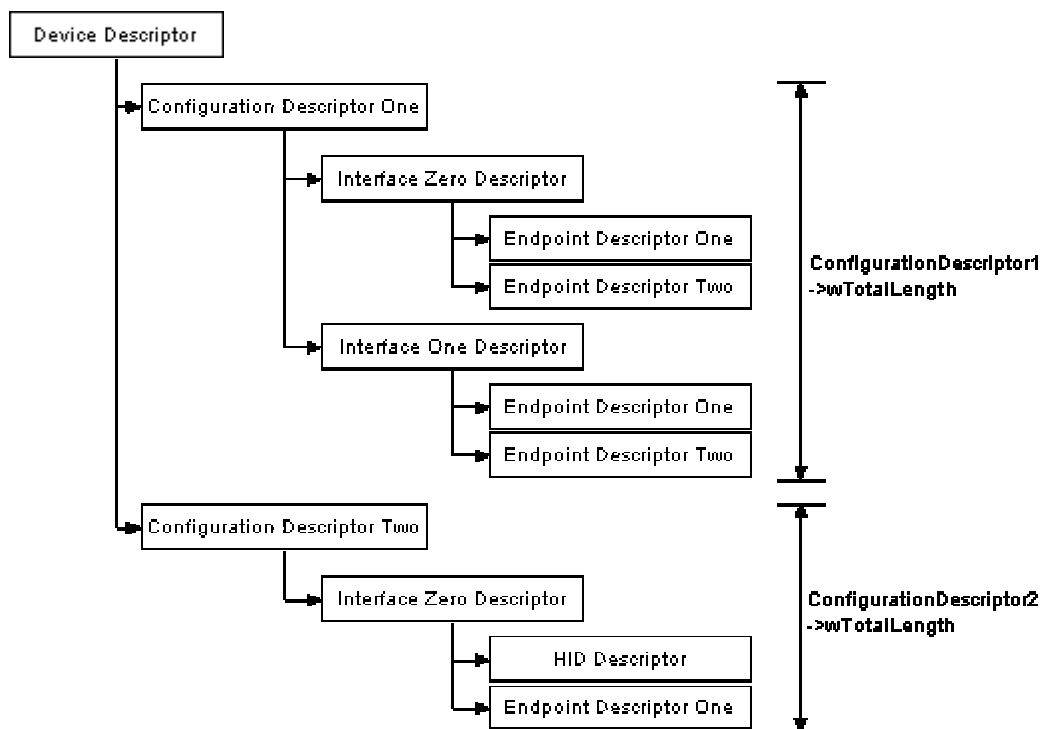
Un appareil USB peut avoir plusieurs configurations différentes alors que la majorité des appareils sont simples et n'en ont qu'une. Le Descripteur de Configuration précise la façon dont l'appareil est alimenté, quelle est sa consommation électrique maximale, le nombre d'interfaces qu'il possède. Il est donc possible d'avoir 2 configurations, quand il est alimenté par le bus et une autre quand il est alimenté par le secteur. Comme ceci est un "en tête " de descripteur d'interface, il est aussi possible d'avoir une configuration utilisant un mode de transfert différent de celui d'une autre configuration.

Une fois que toutes les configurations ont été examinées par l'hôte, celui-ci enverra une instruction **SetConfiguration** avec une valeur différente de Zéro qui correspondra à la valeur **bConfiguration** de l'une des configurations. Elle est utilisée pour sélectionner la configuration voulue.

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du Descripteur en Octets.
1	bDescriptorType	1	Constante	Descripteur de configuration. (0x02)
2	wTotalLength	2	Nombre	Longueur totale en octets de données renvoyées.
4	bNumInterfaces	1	Nombre	Nombre d'Interfaces
5	bConfigurationValue	1	Nombre	Valeur à utiliser comme argument pour sélectionner cette configuration.
6	iConfiguration	1	Index	Index du Descripteur de chaînes de caractères décrivant cette configuration.

7	bmAttributes	1	Bitmap	D7 réservé, mis à 1 ( USB 1.0 alimenté par le bus )  D6 Auto alimenté  D5 Activation d'une station à distance.  D4..0 Réservé. Mis à 0
8	bMaxPower	1	mA	Consommation électrique maximale en unités de 2 m A.

Lors de la lecture du descripteur de configuration, il renvoie la hiérarchie ou l'arborescence complète de configuration qui inclut toute interface apparentée et les descripteurs de terminaisons. Le champ **wTotalLength** indique le nombre d'octets dans la hierarchie.



Le champ **bNumInterfaces** indique le nombre d'interface présent pour cette configuration.

Le champ **bConfigurationValue** est utilisé par la demande SetConfiguration pour sélectionner cette configuration.

Le champ **iConfiguration** est un index de descripteur de chaine de caractère décrivant la configuration dans un format lisible par l'homme.

Le champ **bmAttributes** précise les paramètres d'alimentation pour la configuration. Si un appareil est auto-alimenté, il positionne D6. Le bit D7 était autrefois utilisé par l'USB 1.0 pour indiquer un appareil alimenté par le bus, ceci est maintenant réalisé par le champ **bMaxPower**. Si un appareil utilise l'énergie du bus, que ce soit un appareil alimenté par le bus ou un appareil auto-alimenté, il doit rapporter sa consommation électrique dans le champ **bMaxPower**. Les appareils peuvent aussi prendre en charge l'activation d'une station à distance qui permettra à l'appareil de réveiller l'hôte quand celui-ci est en mode veille.

Le champ **bMaxPower** définit la consommation électrique maximale que l'appareil peut prendre du bus. Elle est donnée en unités de 2 mA, jusqu'au chiffre maximum de 500 mA environ. La spécification permet à un appareil alimenté par un bus de forte puissance de fonctionner jusqu'à 500 mA à partir de Vbus. Dans le cas où un appareil perd son alimentation externe, il ne doit pas fonctionner plus que ce qui est indiqué dans **bMaxPower**. Il devrait échouer sur toutes opérations nécessitant l'alimentation externe.

## Les Descripteurs d'Interfaces

Le Descripteur d'Interface peut être vu comme un " en tête " ou un regroupement de Terminaisons dans un groupe fonctionnel exécutant une simple fonction pour l'appareil. Le Descripteur d'Interface correspond au format suivant :

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du descripteurs en Octets ( 9 octets )
1	bDescriptorType	1	Constante	Descripteur d'interface ( 0x04 )
2	bInterfaceNombre	1	Nombre	Nombre d'Interfaces
3	bAlternateSetting	1	Nombre	Valeur utilisée pour sélectionner une configuration de remplacement.
4	bNumEndpoints	1	Nombre	Nombre de terminaisons utilisées pour cette interface.
5	bInterfaceClass	1	Class	Code Classe ( Assigné par USB org )
6	bInterfaceSubClass	1	SubClass	Code Sous Classe ( Assigné par USB org )
7	bInterfaceProtocol	1	Protocol	Code du Protocole (Assigné par USB org )
8	iInterface	1	Index	Index du descripteur de chaîne décrivant cette interface.

Le champ **bInterfaceNumber** indique l'index du descripteur d'interface. Il doit être pointé à Zéro, et incrémenté une fois pour chaque nouveau descripteur d'interface.

Le champ **bAlternativeSetting** peut être utilisé pour préciser les interfaces de remplacement. Ces interfaces alternatives peuvent être sélectionnées par la demande SetInterface.

**bNumEndpoints** indique le nombre de terminaisons utilisé par l'interface. Cette valeur devrait exclure la terminaison Zéro et est utilisée pour indiquer le nombre de Descripteurs de terminaisons à suivre.

**bInterfaceClass**, **bInterfaceSubClass** et **bInterfaceProtocol** peuvent être utilisés pour préciser les classes prises en compte ( par exemple : HID, Communications, mémoire de masse etc...). Ceci permet à plusieurs appareils d'utiliser des " drivers " (pilote logiciel) de classe évitant le besoin d'écrire des " drivers " spécifiques pour votre appareil.

**iInterface** permet d'avoir une description textuelle de l'interface.

## Les Descripteurs de Terminaisons.

Les Descripteurs de Terminaison sont utilisés pour décrire les terminaisons autres que la terminaison zéro. La terminaison zéro est toujours censée être une terminaison de commande et est configuré avant que n'importe quel autre descripteur ne soit sollicité. L'Hôte utilisera l'information renvoyée par ces descripteurs pour déterminer les besoins de bande passante du Bus.

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du Descripteur en octets ( 7octets ).
1	bDescriptorType	1	Constante	Descripteur de Terminaison (0x05)
2	bEndpointAddress	1	Terminaison	Adresse de Terminaison Bits 0..3b Numéro de terminaison Bits 4..6b Réservés. Mis à 0 Bits 7 Direction ; 0 = Sortie, 1 = Entrée (Ignoré pour les terminaisons de commandes)
3	bmAttributes	1	Bitmap	Bits 0..1 Type de transfert 00 = 00 Commande 01 = Isochrone 10 = par Bloc 11 = Interruption  Bits 2..7 sont réservés. Si terminaison isochrone: Bits 3..2 = Type de synchronisation (Mode Iso) 00 = Pas de synchronisation 01 = Asynchrone 10 = Adaptif 11 = Synchrone  Bits 5..4 = Type d'utilisation (Mode Iso) 00 = Terminaison de données 01 = Terminaison de retour (FeedBack) 10 = Renvoi explicite de terminaison de données 11 = Reservé
4	wMaxPacketSize	2	Nombre	Taille maximale du paquet que cette terminaison est capable d'envoyer ou de recevoir
6	bIntervale	1	Nombre	Intervalle de temps pour interroger les transferts de données de la terminaison. Valeur en nombre de trames. Ignoré pour les terminaisons par Bloc et de commande. Pour le mode Isochrone il doit être égal à 1 et le champ peut valoir de 1 à 255 pour des terminaisons d'interruptions.

**bEndpointAddress** indique quelle terminaison ce descripteur décrit.



**bmAttributes** précise le type de transfert. Cela peut être soit des transferts de type Commande, Interruption, Isochrone ou par Blocs. Si une terminaison Isochrone est précisée, des attributs supplémentaires peuvent être sélectionnés tel que la synchronisation et les types d'utilisations.

**wMaxPacketSize** indique la taille maximale de charge utile pour cette terminaison.

**bInterval** est utilisée pour préciser cet intervalle d'interrogation de certains transferts. L'unité est exprimé en trames équivalent ainsi à 1 ms pour des appareils basse / pleine vitesse et 125 µs pour des appareils haute vitesse.

### Les Descripteurs de chaînes de caractères.

Les Descripteurs de chaînes fournissent une information lisible pour l'homme et sont optionnels. S'ils ne sont pas utilisés, tout champ d'index de descripteurs de chaînes doit être mis à zéro indiquant qu'il n'y a pas de descripteur de chaîne disponible.

Les chaînes de caractères sont codées au format Unicode et les produits peuvent être prévus pour comprendre les diverses langues. L'index de chaîne zéro devra retourner une liste de langues acceptée. On peut trouver une liste USB d'identification de langue dans Universal Serial Bus Language Identifiers (LANGIDs) version 1.0 ( Identificateurs de langue sur BUS Série Universel version 1.0)

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du descripteur en octets
1	bDescriptorType	1	Constante	Descripteur de chaîne (0x03)
2	wLANGID[0]	2	Nombre	Langue acceptée code zéro ( par exemple 0x0409 Anglais, U.S.)
4	wLANGID[1]	2	Nombre	Langue acceptée code Un ( par exemple 0x0C09 Anglais, Australien)
n	wLANGID[x]	2	Nombre	SLangue acceptée code x ( par exemple 0x0407 Allemand - Standard)

Le descripteur de chaînes ci-dessus montre le format du descripteur de chaîne zéro. L'Hôte devra lire ce descripteur pour déterminer quelles langues sont disponibles. Si une langue est acceptée, elle peut être référencée en envoyant l'identification de la langue dans le champ **wIndex** à la demande de Get\_Descriptor(String).

Toutes les chaînes de caractères à venir tiennent dans le format ci-dessous :

Décalage	Champ	Taille	Valeur	Description
0	bLength	1	Nombre	Taille du descripteur en octets
1	bDescriptorType	1	Constante	Descripteur de chaînes (0x03)
2	bString	n	Unicode	Textes codés unicode